

Dense, Efficient Chip-to-Chip Communication at the Extremes of Computing

Thesis by

Matthew Loh

In Partial Fulfilment of the Requirements

for the Degree of

Doctor of Philosophy



CALIFORNIA INSTITUTE OF TECHNOLOGY

Pasadena, California

2013

(Defended 03 May 2013)

©2013

Matthew Loh

All Rights Reserved

S.D.G.

Acknowledgements

Although only one person graduates as a direct result of this dissertation, there are a great many without whose efforts, great and small, its completion would have been impossible.

First among these is my advisor, Prof. Azita Emami-Neyestanak. Being the very first student to join her group, I have been impressed by her dedication to excellence both as a researcher as well as a teacher and mentor, and how she has developed as both during my time as her student. Her support, advice and encouragement have been a defining and essential part of my journey through graduate school.

I would like to thank the members of my candidacy and defense committees, Prof. Ali Hajimiri, Prof. Yu-Chong Tai, Prof. Dave Rutledge and Dr. Sander Weinreb, for their willingness to participate in and evaluate my research, and for their probing questions and valuable input.

The beginning of graduate studies places you at the bottom of a steep and daunting learning curve; not least among the challenges is figuring out how to build a test-setup for the chips you have so painstakingly designed. For patiently helping me to learn the ins-and-outs of high-speed PCB design and wirebonding, and providing much advice on (and loans of) test equipment, I am deeply indebted to Dr. Sander Weinreb, Hamdi Mani, Steve Smith and Hector Ramirez.

I would also like to express my gratitude to the students and post-doctoral scholars of the Mixed-Signal Integrated Circuits and Systems (MICS) and Caltech High-speed Integrated Circuits (CHIC) groups at Caltech; in no particular order, Juhwan Yoo, Meisam Hornarvar Nazari, Mayank Raj, Manuel Monge, Saman Saaedi, Krishna Settaluri, Kaveh Hosseini, Steve Bowers, Ed Keehr, Kaushik Sengupta, Florian Bohn, Aydin Babakhani, Hua Wang, Kaushik Dasgupta, Alex Pai, Behrooz Abiri, Constantine Sideris, Amir Safaripour and Firooz Aflatouni. The positive atmosphere that exists on the third floor of Moore is sustained by their dedication, both to research itself as well as to each other as friends and colleagues. Much that I have accomplished through my time as a graduate student would have been more painful or even impossible without their encouragement and advice. I would especially like to thank Juhwan Yoo,

who joined me in the MICS group almost at its very beginning, provided an invaluable partner to hash out ideas with, and was an unimpeachable apartment-mate for two years.

The smooth operation of the lab, building and department would be impossible without the efforts of people such as the group administrator, Michelle Chen, the option coordinator, Tanya Owen, the department administrator, Carol Sosnowski, and the building engineer/manager/go-to-guy, Kent Potter. Their assistance was essential at all stages of my studies at Caltech, and I owe them much for their patience and support. I am also indebted for the computer and network support provided by our IMSS staff, Gary Waters, John Lilley, Chris Birtja and Dan Caballero.

Research in IC design is complex and expensive, enabled in no small part thanks to the companies and funding agencies that support it; I would like to thank ST Microelectronics and IBM for chip fabrication, and Intel, the NSF and the C2S2 Focus Center for funding support.

Life at Caltech would not have been complete without the many undergraduates with whom I crossed paths – whether as a resident of Avery House, teaching assistant for EE 45 or summer research co-mentor. Though this list is by no means exhaustive, I would like to thank Karthik Sarma, Chris White, Chris Wong, James Jester, Joseph Schmitz, Ryan Hammerly, Brandon Hensley, Sam Elder, Dan Thai, Angie Wang, Christina Lee, John Liu, David Lee, Cherrie Soetjipto, David Hu, Po-Ling Loh, Christian Griset, Nina Ng-Quinn, Brian Peng and Casey Glick. By offering a refreshing perspective on academics, science and research, and injecting into my life enthusiasm and verve that is frequently drained away by the demands of graduate study, their friendship helped sustain and inspire me throughout my time here.

Throughout the course of my graduate studies, I have received a tremendous amount of support from my parents, Andrew and Li, and my sister, Amanda. I thank them for their love and prayers, which have been a great encouragement to me despite the distance that separates us.

None of this could have happened but for the grace of God. In the face of setbacks and triumphs, whether I am enthusiastic or jaded and indifferent, stressed out and frustrated or calm and relaxed, it is by His providence alone that I am sustained each day.

My wife, Joy, deserves no end of kudos for sticking with me through this whole process, giving her support and love even when I deserved them least, and for patiently putting up with my non-committal answers to the age-old question: how long more? I now, finally, have an answer.

Abstract

The scalability of CMOS technology has driven computation into a diverse range of applications across the power consumption, performance and size spectra. Communication is a necessary adjunct to computation, and whether this is to push data from node-to-node in a high-performance computing cluster or from the receiver of wireless link to a neural stimulator in a biomedical implant, interconnect can take up a significant portion of the overall system power budget. Although a single interconnect methodology cannot address such a broad range of systems efficiently, there are a number of key design concepts that enable good interconnect design in the age of highly-scaled CMOS: an emphasis on highly-digital approaches to solving ‘analog’ problems, hardware sharing between links as well as between different functions (such as equalization and synchronization) in the same link, and adaptive hardware that changes its operating parameters to mitigate not only variation in the fabrication of the link, but also link conditions that change over time. These concepts are demonstrated through the use of two design examples, at the extremes of the power and performance spectra.

A novel all-digital clock and data recovery technique for high-performance, high density interconnect has been developed. Two independently adjustable clock phases are generated from a delay line calibrated to 2 UI. One clock phase is placed in the middle of the eye to recover the data, while the other is swept across the delay line. The samples produced by the two clocks are compared to generate eye information, which is used to determine the best phase for data recovery. The functions of the two clocks are swapped after the data phase is updated; this ping-pong action allows an infinite delay range without the use of a PLL or DLL. The scheme's generalized sampling and retiming architecture is used in a sharing technique that saves power and area in high-density interconnect. The eye information generated is also useful for tuning an adaptive equalizer, circumventing the need for dedicated adaptation hardware.

On the other side of the performance/power spectra, a capacitive proximity interconnect has been developed to support 3D integration of biomedical implants. In order to integrate more functionality while staying within size limits, implant electronics can be embedded onto a foldable parylene (‘origami’) substrate. Many of the ICs in an origami implant will be placed face-to-face

with each other, so wireless proximity interconnect can be used to increase communication density while decreasing implant size, as well as facilitate a modular approach to implant design, where pre-fabricated parylene-and-IC modules are assembled together on-demand to make custom implants. Such an interconnect needs to be able to sense and adapt to changes in alignment. The proposed array uses a TDC-like structure to realize both communication and alignment sensing within the same set of plates, increasing communication density and eliminating the need to infer link quality from a separate alignment block. In order to distinguish the communication plates from the nearby ground plane, a stimulus is applied to the transmitter plate, which is rectified at the receiver to bias a delay generation block. This delay is in turn converted into a digital word using a TDC, providing alignment information.

Contents

Acknowledgements	iv
Abstract	vi
Contents.....	viii
List of Figures	xi
List of Tables.....	xvii
Chapter 1: Introduction.....	1
1.1 Interconnect in digital systems	2
1.2 High-bandwidth/high-power systems.....	3
1.3 Low-bandwidth/low-power systems	4
1.4 Organization.....	6
Chapter 2: High-Speed Electrical Interconnect.....	8
2.1 Clocking	9
2.1.1 Sub-rate clocking.....	11
2.2 Clock and data recovery	12
2.2.1 Phase detectors for 2x oversampled CDR.....	14
2.2.2 Phase detectors for baud rate CDR	17
2.2.3 Clock generation	19
2.3 Equalization	23
2.3.1 Transmitter pre-emphasis	27
2.3.2 Receiver linear equalization	28
2.3.3 Decision-feedback equalization.....	29
2.4 Summary	30

Chapter 3:	All-Digital Clock and Data Recovery	32
3.1	Ping-Pong CDR	34
3.1.1	Search algorithm	37
3.1.2	Search filtering	38
3.1.3	System startup and corner cases.....	44
3.2	Shared CDR.....	45
3.3	Implementation	46
3.3.1	Phase generator.....	48
3.3.2	Phase generator linearity	50
3.3.3	Multiplexers	53
3.3.4	Retiming logic	55
3.4	Hardware measurements	56
3.5	Equalization adaptation.....	64
3.5.1	On-chip links.....	64
3.5.2	Eye-monitor-based adaptive equalization	70
3.6	Summary.....	76
Chapter 4:	Proximity Communication.....	77
4.1	Capacitive Proximity Interconnect	79
4.1.1	Transmitter and receiver design	81
4.1.2	Sensing and adapting to misalignment	84
4.2	Inductive Proximity Interconnect	89
4.2.1	Transmitter and receiver design	94
4.3	Proximity Interconnect for Low-power Systems.....	97
Chapter 5:	Capacitive Proximity Communication for Origami Implants.....	101

5.1	Plate and array design	102
5.2	Transceiver array with distributed alignment sensing.....	108
5.2.1	Alignment sensing	109
5.2.2	Transmitter and receiver.....	117
5.3	Hardware measurements	121
5.4	Summary	129
Chapter 6:	Conclusion	131
	List of Abbreviations	134
	Bibliography	136

List of Figures

Figure 1.1: Chip-to-chip interconnect in a typical 2-socket server. Dashed lines outline individual CPU chips. Each link shown operates at >1 Gb/s.	3
Figure 1.2: Effect of degenerative retinal diseases.	5
Figure 2.1: Fundamental clocked link. The transmitter (Tx) sends data ($x[n]$) through channel, which the receiver (Rx) converts into an estimate ($\hat{x}[n]$).	8
Figure 2.2: Eye diagram, showing sources of noise and timing and voltage margins.	9
Figure 2.3: Source-synchronous link with shared timing recovery.	10
Figure 2.4: Plesiochronous link.	11
Figure 2.5: Quarter-rate link.	12
Figure 2.6: Generalized 2x oversampled CDR.	13
Figure 2.7: Generalized baud rate CDR.	14
Figure 2.8: Hogge phase detector and timing diagram, showing integrated output gradually rising in response to phase difference between data and clock.	15
Figure 2.9: Alexander phase detector, showing sample locations when clock leads or lags.	16
Figure 2.10: Example impulse response (adapted from [30]), showing the sample points of converged Mueller-Müller CDR, comparing Type A (red) and Type B (green). Sample points are UI spaced.	17
Figure 2.11: Architecture of Mueller-Müller Type A CDR.	18
Figure 2.12: Open-loop delay line with 4 delay elements.	19
Figure 2.13: Phase interpolator, with weighted buffers ($\alpha < 1$). Output timing shown in ideal linear case.	20
Figure 2.14: (a) PLL block diagram, (b) phase-domain model and (c) typical loop filter.	20
Figure 2.15: (a) DLL block diagram and (b) phase domain model.	22
Figure 2.16: Pulse response of a legacy 16" server backplane at 12 Gb/s, showing sample points and the effects of dispersion and reflection.	24
Figure 2.17: Frequency response of legacy 16" server backplane [37].	24
Figure 2.18: Typical server backplane, with major sources of reflections marked.	26

Figure 2.19: Via (a) before and (b) after back-drilling, showing effect on S_{21} (of the stub only)...	26
Figure 2.20: Discrete-time (FIR) transmitter pre-emphasis with i pre-cursor taps and j post-cursor taps.....	28
Figure 2.21: Receiver with linear equalizer and DFE.	29
Figure 3.1: Single-pin ping-pong CDR architecture overview.....	34
Figure 3.2: Ping-pong CDR algorithm.	35
Figure 3.3: Steps in a UI swap process. Even (rising) and odd (falling) edges of the clocks marked.	36
Figure 3.4: Search procedure, showing movement of search clock and numbered eye edges.	37
Figure 3.5: CDR operation with eye drifting until UI swap is required. Relevant eye edges are marked and numbered.....	38
Figure 3.6: Eye information collection via mismatch counter and AND/OR filter.	39
Figure 3.7: Typical BER bathtub, and the probability of mismatch declaration at each phase position with ($nbase = 32$) and without ($n = 32$) repeated averaging.....	41
Figure 3.8: Probability of match-to-mismatch transition detection, $n = nbase = 32$	42
Figure 3.9: Peak probability in distribution of match/mismatch transition detection, with and without repeated averaging.....	43
Figure 3.10: Probability of match-to-mismatch transition detection, $n = nbase = 128$	43
Figure 3.11: AND/OR filter with $k = 4$	44
Figure 3.12: Sharing concept and modified draft algorithm, for 3 data pins and 4 clocks	45
Figure 3.13: Three-pin shared CDR architecture.	47
Figure 3.14: Phase generator architecture.....	48
Figure 3.15: Delay line with (a) delay cell and (b) phase interpolator. Weak cross-coupled inverters are marked with a 'W'.	48
Figure 3.16: Delay line with negative INL and phase interpolator with (a) negative INL, (b) positive INL and (c) 'crossing' INL.....	52
Figure 3.17: Search/data multiplexer (only clock routing shown; sample routing is similar).....	53
Figure 3.18: Retiming logic.	55
Figure 3.19: Die micrograph and core detail.....	56

Figure 3.20: Data for 2 UI of delay, over the delay line calibration code range.	57
Figure 3.21: Phase generator non-linearity.	58
Figure 3.22: Delay line output non-linearity (note that the 0 th output is missing, since this is inaccessible due to the configuration of the phase interpolator).....	58
Figure 3.23: Interpolator INL, across different groups of 8 phase positions (each corresponding to one sweep through the interpolator).....	59
Figure 3.24: Interpolator DNL, across different groups of 8 phase positions (each corresponding to one sweep through the interpolator).....	59
Figure 3.25: SJ tolerance with control logic clock at 40 MHz, for 3 x 9 Gb/s PRBS-7 input and $\text{BER} < 10^{-12}$	60
Figure 3.26: (a) Frequency offset tolerance scaling for 3 x 9 Gb/s PRBS-7 input (measured $\text{BER} < 10^{-12}$ and simulated $\text{BER} < 10^{-6}$), with (b) low frequency detail.	61
Figure 3.27: Effect of n_{base} on frequency offset tolerance, simulated on a single channel at 625 MHz with PRBS-7 input and $\text{BER} < 10^{-6}$	61
Figure 3.28: Power breakdown and scaling performance	62
Figure 3.29: On-chip links, within the context of a two-socket server. Individual CPU chips indicated by dashed lines. (a) repeated, full-swing and (b) RC-limited, low-swing links shown.....	64
Figure 3.30: Bandwidth density (Hz/ μm) estimated over wire pitches up to 3 μm in metal 7 of a typical 9-metal 65 nm process.....	67
Figure 3.31: Bandwidth density-optimized on-chip channel in typical 9-metal 65 nm process, with frequency response for 10 mm length.....	68
Figure 3.32: Pulse response of channel in Figure 3.31, at 5 Gb/s.	68
Figure 3.33: (a) Classic DFE with discrete-time FIR feedback, and (b) DFE with continuous- time RC feedback, replacing all N taps in the FIR feedback.....	69
Figure 3.34: Simulation of ping-pong CDR-based adaptive equalizer, showing stages of operation.	74
Figure 3.35: Histograms of received signal level at sample point, comparing continuous-time IIR DFE with ping-pong CDR-based adaptation @ 5 Gb/s (red) against 7-tap FIR DFE with SS-LMS adaptation @ 3.25 Gb/s (green)	75
Figure 4.1: Basic structure of capacitive proximity interconnect, with equivalent circuit	79

Figure 4.2: Capacitive link, showing transmit driver resistance and receiver bias resistance.....	81
Figure 4.3: Variable-threshold comparator used as a receiver in a capacitive proximity interconnect, with waveforms showing principle of operation	82
Figure 4.4: Example precharge-and-evaluate input stage (adapted from [75]). During precharge (clock is high), bias point is set by shorting the input and output of the input inverter stage, causing it to enter its metastable state. The D flip-flop evaluates the current bit at the rising edge of the clock.....	83
Figure 4.5: High R_{bias} achieved using leakage device	83
Figure 4.6: Chip-to-chip alignment can be described in terms of 6 axes: 3 linear (x, y and z) and 3 angular (θ_x , θ_y , θ_z).....	84
Figure 4.7: Vernier bar-based alignment sensor [9]. Output of flip-flop depends on the most strongly coupled transmitter bar, and can be unknown ('X') if two adjacent transmitter bars are equally-well coupled	85
Figure 4.8: Multi-plate alignment sensor with passive target chip [79], for (a) in-plane (x- and y-axis) alignment and (b) vertical (z-axis) alignment, showing equivalent circuits...	86
Figure 4.9: Ring oscillator-based alignment sensor [80]	87
Figure 4.10: Adaptation of transmitter array to receiver array alignment [81]	88
Figure 4.11: Circular wire loops for calculation of mutual inductance	89
Figure 4.12: Comparison of mutual inductance between two single-loop wires with $r_1 = r_2 = 50$ μm and capacitance between two $50 \times 50 \mu\text{m}$ square parallel plates in silicon dioxide ($\epsilon_r = 3.9$), over different amounts of separation.....	90
Figure 4.13: Coupling co-efficient between two single-loop wires with $r_1 = r_2 = 50 \mu\text{m}$, over different amounts of separation.....	91
Figure 4.14: Ideal inductively-coupled link, with equivalent circuit	91
Figure 4.15: Inductively-coupled link with parasitics.....	92
Figure 4.16: Simplified inductively-coupled link model, using a unilateral coupled inductor	92
Figure 4.17: Magnitude response of circuit with full parasitic model, compared with simplified circuit using unilateralized coupled inductor model, with $L_1 = L_2 = 5 \text{ nH}$, $k =$ 0.2 , $R_{p1} = R_{p2} = 100 \Omega$, $R_{tx} = 10 \text{ k}\Omega$, $C_{p1} = 75 \text{ fF}$ and $C_{p2} = 100 \text{ fF}$	94
Figure 4.18: Constant-current transmitters: (a) H-bridge, (b) CML-based and (c) operation waveforms.....	94

Figure 4.19: Variable-threshold comparators: (a) CML-based, (b) inverter-based and (c) operation waveforms	95
Figure 4.20: Pulsed-current transmitters: (a) H-bridge with delay line, (b) single-ended with storage capacitor and (c) operation waveforms, showing small timing margin available at receiver	96
Figure 4.21: Survey of wearable and implantable biomedical devices reported in ISSCC between 2010 and 2013	98
Figure 5.1: Top-down view of the sensor array, in (a) best-case (maximum overlap) and (b) worst-case (minimum overlap) alignment. Active sensor plates are shaded in green, and the associated target plate is outlined.....	103
Figure 5.2: Plate and switch configurations for (a) $n = 2$, (b) $n = 3$ and (c) $n = 4$. The highlighted group of active sensor plates indicates the worst-case loading condition, where the largest number of inactive switches are connected to the active plates	104
Figure 5.3: Link gain for different values of n , when plates are in best-case alignment	106
Figure 5.4: Link gain for different values of n , when plates are in worst-case alignment.....	106
Figure 5.5: Dielectric and metal layers used to form plate structure, and sensor/target array arrangement (target chip outline not shown for clarity)	107
Figure 5.6: Architecture of the sensor and target cells, with key functional blocks indicated.....	108
Figure 5.7: (a) Standard TDC using a single delay line compared to (b) a vernier TDC. Flip-flops act as arbiters, indicating which edge arrives earlier	110
Figure 5.8: Sensor array structure, showing TDC path for alignment sensing at indicated plates	110
Figure 5.9: Rectifier and associated timing diagram.....	111
Figure 5.10: (a) Differential voltage-controlled delay line with variable-threshold output buffer and (b) variable-delay inverter used in VCDL unit cell	112
Figure 5.11: TDC delay cell and arbiter	113
Figure 5.12: Target-to-sensor link gain vs. air gap, using $n = 2$, target plate size of $60 \times 60 \mu\text{m}$ and no parylene.....	114
Figure 5.13: Target-to-sensor link gain vs. air gap, using $n = 2$, target plate size of $60 \times 60 \mu\text{m}$ and $12 \mu\text{m}$ parylene.....	114

Figure 5.14: Two adjacent groups of sensor plates used for x-axis alignment sensing. $0 \leq m \leq 1$; when $m = 0$, target plate is all the way to the left (completely over $V1$). Likewise, when $m = 1$, target plate is completely over $V2$	115
Figure 5.15: In-plane alignment estimation, using simulation results with various dielectrics between the two chips	116
Figure 5.16: Equivalent circuit of the capacitive link, with switch parasitics (R_{sw} and C_{sw}) introduced. R_{bias} is assumed to be very large, and is omitted	117
Figure 5.17: Tri-state buffer-based transmitter, with leakage path to define plate bias voltage ..	118
Figure 5.18: Source-follower buffer with gateable Wilson current mirror bias	118
Figure 5.19: 3-stage hybrid low-pass filter	119
Figure 5.20: Input slicer and offset compensation (SR latch not shown). ‘Reset’ zeroes the offset compensation capacitor and ‘oc_en’ is asserted during offset compensation calibration	120
Figure 5.21: Input slicer offset estimated across 200 Monte Carlo simulation runs, (a) before and (b) after offset compensation	121
Figure 5.22: Die micrograph, with sensor and target arrays marked	121
Figure 5.23: Sensor and target cell layout detail	122
Figure 5.24: Test setup. Inset: Detail of chips when brought into alignment	123
Figure 5.25: Effect of offset compensation on a single sensor cell's VCDL/TDC-based ADC	124
Figure 5.26: The effect of VCDL/TDC offset compensation on (a) offset error and (b) total error, measured over 144 sensor cells across 6 chips	125
Figure 5.27: Alignment sensor output under vertical (z-axis) separation	125
Figure 5.28: Comparison of measured and simulated (quantized and unquantized) in-plane (x- and y-axis) alignment sensing, with (a) air-only and (b) 2x6 μm parylene dielectrics	126
Figure 5.29: Alignment sensor output under in-plane (x- and y-axis) misalignment, for 4, 5, 6, 2x4 and 2x5 μm parylene dielectrics	127
Figure 5.30: Achieved in-plane alignment sensor resolution vs. parylene dielectric thickness	127
Figure 5.31: Maximum data rates achievable ($\text{BER} < 10^{-9}$) under best-case alignment, for various thickness of parylene dielectric	129

List of Tables

Table 1.1: Approximate chip-to-chip I/O bandwidths, <i>total</i> system power, link distance and form-factor for different system types.....	2
Table 2.1: Truth table for Alexander phase detector. ‘X’ states indicate don’t cares, since $X \neq Y$ cannot normally co-exist with $X=Z$	16
Table 3.1: Performance summary, 3 x 9 Gb/s ping-pong CDR.....	63
Table 5.1: Number of inactive switches loading the active group of plates, for various values of n	105
Table 5.2: Performance summary, 12 x 60 Mbps capacitive proximity interconnect with embedded alignment sensor	128

Chapter 1: Introduction

In observing that the economically-efficient number of components per integrated circuit (IC) had increased 2x every year between 1962 and 1965 and predicting that this trend could be sustained, Moore’s seminal 1965 paper [1] provided an impetus and objective for what is, perhaps, the greatest economic engine in history. Certainly it has sparked a revolution in computation and communication whose full effects are only beginning to be understood.

Integration of more and smaller transistors has allowed increasing complexity in the design of processing and communication units, while driving (at least initially) a rise in clock speeds and power efficiency. Even a slowdown in voltage scaling and the advent of power consumption as a major limiter in clock speed scaling have not prevented continued advances in computational power; increases in transistor density have enabled designers to work around these limits by adopting new approaches such as multi-core and heterogeneous computing. A good example of this is the incorporation of vector processors, such as graphical processing units (GPUs), into a general-purpose computational architecture.

The economics of CMOS scaling have also pushed ICs into new and unusual spaces. For example, the availability of very cheap, highly-integrated microcontrollers has spurred a renaissance in hobbyist electronics and led to innovation in ubiquitous computing – the idea that electronics can be made cheap and small enough to afford a measure of ‘intelligence’ and interactivity to even the most mundane of objects. In practice, this has enabled a range of projects from wearable electronics [2] to sophisticated, home-built 3D printers [3]. The versatility of CMOS has also made it an attractive platform for implementing sensing, stimulus, processing and communication systems in biomedical implants. The high level of integration possible makes it particularly attractive in this space, where power consumption and package size are of primary importance.

1.1 Interconnect in digital systems

The power of scaling, then, has created a strong economic incentive to apply CMOS technology in a diverse range of applications across the power consumption, performance and size spectra. Of course, where there is computation, a need exists to communicate with and between the elements realizing that computation. Whether this is to facilitate pushing data from node-to-node in a high-performance computing (HPC) cluster, between CPU and memory in a desktop, or from the receiver of wireless link to a neural stimulator in a biomedical implant, interconnect can take up a significant portion of the overall system power budget (e.g. $\sim 11\%$ for off-chip interconnect in the Intel Nehalem-EX microprocessor [4]). As a result, considerable design effort is expended to ensure that interconnect blocks operate as efficiently as possible, while still meeting performance and robustness targets.

System	I/O Bandwidth (Gb/s)	System Power (W)	Link distance	Form-factor (volume, L)
Server/workstation	>2000	>300	>10 cm	>60
Desktop	200-300	60-100	5-30 cm	20-40
Laptop	190-270	20-35	5-20 cm	1-4
Tablet	100	10	5-10 cm	0.25-0.5
Smartphone	40	<1	1-10 cm	0.07-0.12
Biomedical implant	<0.1	<0.1	<1 cm	<0.01

Table 1.1: Approximate chip-to-chip I/O bandwidths, *total* system power, link distance and form-factor for different system types.

Table 1.1 gives a sense of the scale of the design problem at the time of writing; depending on the target application, chip-to-chip I/O bandwidths range over 4 orders of magnitude, and system power consumption and form-factor over 3 orders of magnitude or more. Since a single interconnect methodology cannot address such a broad range of requirements efficiently, chip-to-chip communication systems are typically customized to specific applications. Nevertheless, there are a number of design concepts that typify efficient interconnect design, no matter what the application; these concepts are elucidated in this dissertation.

1.2 High-bandwidth/high-power systems

High-performance multi-processor/multi-socket systems, typically used in the server and HPC space, require many high-speed communication links (Figure 1.1). Whether these are between processors, to system memory or to high-performance add-in modules (such as GPUs), the fundamental challenge is the same: realizing multi-Gb/s links within strict power budgets, in the presence of strong attenuation and crosstalk in the channel.

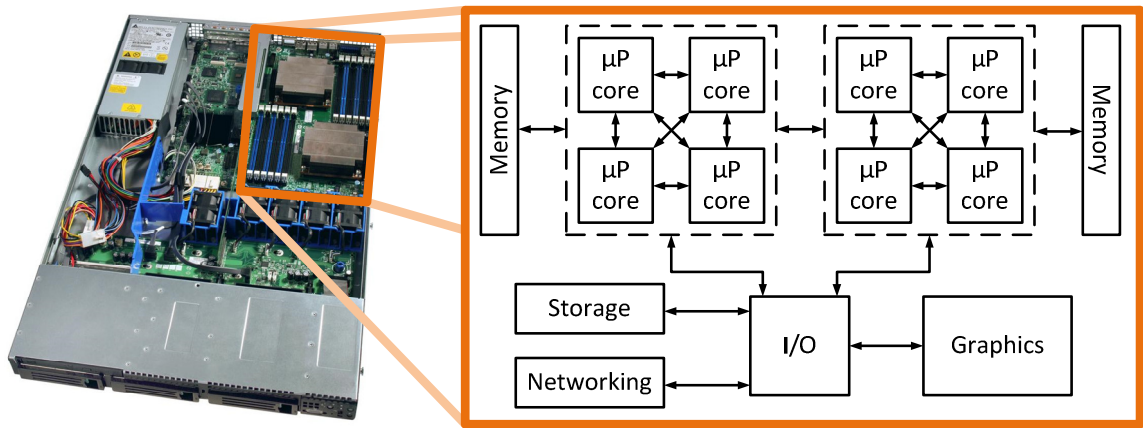


Figure 1.1: Chip-to-chip interconnect in a typical 2-socket server. Dashed lines outline individual CPU chips. Each link shown operates at >1 Gb/s.

Historical trends are not promising – a survey conducted in 2009 showed that, while link power efficiency has improved at a rate of about 20% per year, the demand for bandwidth has outstripped this improvement, increasing between 2 to 3x annually [5]. The same survey suggested that, in order to meet the requirement for multi-Tb/s operation, link energy efficiencies would have to improve an order of magnitude, from 10 pJ/b to 1 pJ/b. To achieve such a target fast enough to meet demand, business-as-usual improvements, which typically rely on the effects of CMOS scaling, are inadequate. Indeed, because chip-to-chip communication is fundamentally a mixed-signal problem (converting a noisy analog waveform into a reliable digital bit-stream) traditional approaches to solving it have relied heavily on analog techniques. As CMOS technology scaling is driven largely by the demand for more and faster digital switches, it is precisely the analog blocks that suffer most from the transition to finer-geometry processes. Therefore, it is necessary to fundamentally re-think traditional approaches to the problems of link

design, with an emphasis on more digital-like solutions. Concurrently, it is important to pursue higher-level optimizations such as sharing hardware, not only between links, but also between functions such as synchronization and equalization.

1.3 Low-bandwidth/low-power systems

If there is an incentive to pursue unconventional approaches to link design in order to address the challenges faced in high-performance systems, this is even more acutely felt as ICs start to address non-traditional, power- and form-factor-sensitive arenas such as ubiquitous computing and biomedical devices.

Although they address very different problems, the intent of both biomedical and ubiquitous computing devices is frequently the same: sense the environment, perform some simple computation to understand the resulting data and respond appropriately. A typical system, then, will include one or more sensors, a processor of some sort (frequently optimized for low power, small size and low cost), a set of actuators and a wireless interface. A battery might also be necessary if power is not delivered wirelessly or scavenged from the environment.

Many of these applications are specialized and low-volume. Producing unique systems-on-chip (SoCs) for each of them is prohibitively expensive, and existing printed circuit board (PCB) fabrication techniques simply cannot achieve the compactness required. Proximity communication [6] provides a means to overcome these limitations – inductive coils (e.g. [7], [8]) or capacitive plates (e.g. [9], [10]) can be fabricated in the top-level metal of a standard CMOS process, with no extra fabrication steps. When these coils or plates are placed in close proximity with each other, inductive or capacitive coupling forms an electrical link that can transfer data and/or power. Since it uses the existing metal stack, proximity communication involves minimal extra cost and enables manufacturing techniques such as Brick-and-Mortar [11], which envisions the mass production of a library of IC sub-blocks (‘bricks’) that can be placed on a standardised communications substrate (‘mortar’). By customizing the mix of bricks used, specialized systems can be built up for particular applications in cost-effective fashion, while still retaining a large part of the compactness, power efficiency and performance advantages of traditional SoC design.

To make this discussion more concrete, consider a retinal prosthesis for treating diseases such as retinitis pigmentosa (RP, a degenerative eye disease that gradually reduces a person's field of vision. Eventually, they can become effectively blind) and age-related macular degeneration (AMD, where the macula, or central portion of the retina, is damaged, resulting in loss of visual fidelity. Again, this can result in clinical blindness) [12].



Figure 1.2: Effect of degenerative retinal diseases.¹

Very few options exist for treating late-stage RP or AMD. In both diseases, although the photoreceptors in the retina have ceased to function properly, the nerves in the retina are still healthy. Therefore, retinal prostheses can be used to electrically stimulate the still-functioning nerves with visual data, restoring partial vision. Although early devices have been successfully implanted, they have suffered limitations in the amount of stimulation current able to be delivered [13] or the number of electrodes (just 16 in [14]). Since visual fidelity is closely related to the number of electrodes that successfully stimulate the retinal neurons, scaling these systems to thousands of electrodes is an important (and challenging) task. However, high voltages ($\sim 5\text{-}10$ V) are necessary to drive the desired stimulus currents, preventing the use of finer-geometry CMOS technologies, which have insufficiently large breakdown voltages. As a result, die size (e.g. $8 \times 8 \text{ mm}^2$ for a proposed 1024-electrode array [15]) prevents a straightforward increase in the number of electrodes, since the prosthesis is intended for implantation in the eye and the incision size is limited.

In order to work around this size limit, large multi-electrode systems can be split into multiple chips and placed on a flexible, bio-compatible substrate, such as Parylene-C [16], which

¹ From National Institutes of Health, National Eye Institute. Available: <http://www.nei.nih.gov/photo/sims/index.asp>. [Accessed: 19 Jun 2012]

can be folded compactly for implantation. By engineering the folding (‘origami’) of the parylene, it can be made to assume useful shapes once deployed (e.g. an inductive coil for power/data delivery [17]). A longer-term objective would be to build up a library of standard and useful ICs (e.g. neural stimulator driver, wireless data/power management, low-power processor) which are compatible with parylene integration, so application-specific implants can be put together from pre-fabricated, mass-produced modules in a manner similar to that proposed by the Brick-and-Mortar scheme mentioned above.

Indeed, the analogy to Brick-and-Mortar is an apt one, since similar chip-to-chip communication techniques are appealing when building a modular origami parylene system. Many ICs in an origami implant will be placed face-to-face with each other when it is deployed, facilitating the use of proximity communication. The ability to communicate wirelessly helps to reduce wired communication density, making it easier to develop more compact implants. Additionally, wireless module-to-module communication eases the assembly of custom implants from origami-based library blocks. As a result, proximity communication (whether via capacitive or inductive coupling) is a promising avenue of research for chip-to-chip interconnect in low-power systems such as this.

1.4 Organization

This dissertation is composed of two major parts, corresponding to the two extremes of digital system interconnect described in the introduction. The first addresses high-speed electrical interconnect: Chapter 2 provides an overview of the terminology and design techniques used in this domain, including techniques for synchronization and clock recovery as well as equalization. Chapter 3 builds on this background and presents a novel all-digital ‘ping-pong’ clock and data recovery system, which is extended for use in an adaptive equalizer. The second half of this dissertation addresses electrical interconnect in the context of low-power, space-constrained systems by using the example of proximity interconnect for origami implants. Chapter 4 describes the development of capacitively and inductively-coupled wireless communication from the very earliest days when it competed with radio (‘Hertzian wave’) communication to present-day

approaches for chip-to-chip and chip-to-package high-speed interconnect. Design considerations for proximity communication in low-power systems is discussed, and Chapter 5 details an implementation of just such a system for use in origami implants. Throughout, it is sought to demonstrate the importance of a more-digital, adaptive, shared and multi-functional hardware approach to efficient design across the extremes of chip-to-chip communication systems.

Chapter 2: High-Speed Electrical Interconnect

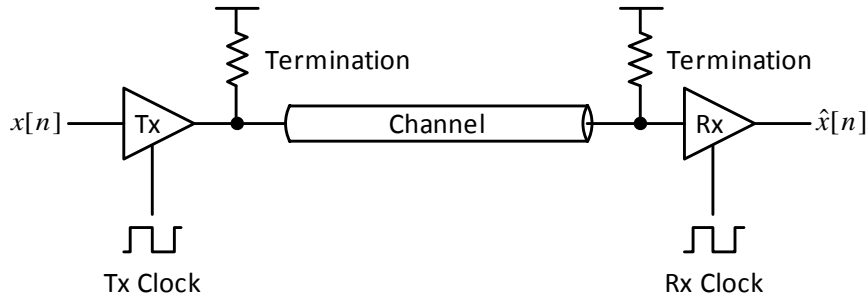


Figure 2.1: Fundamental clocked link. The transmitter (Tx) sends data ($x[n]$) through channel, which the receiver (Rx) converts into an estimate ($\hat{x}[n]$).

Figure 2.1 shows the components and configuration of the most basic clocked electrical link: a transmitter, receiver and channel, with clocks and termination resistors at each end to time the data and reduce the effect of reflections due to impedance discontinuities. Since the signal sent down the channel exists in the continuous time, analog domain, the purpose of the receiver is to determine the optimum decision point, in time and amplitude, to estimate the original bit-stream and minimize errors. In an additive white Gaussian noise (AWGN) channel, the bit error rate (BER) is classically characterized by the voltage margin, V_M , at the sampling point [18]:

$$BER = \exp\left(-\frac{\left(V_M/V_R\right)^2}{2}\right) \quad (2.1)$$

where V_R is the root-mean square (RMS) voltage noise; since Gaussian noise is assumed, this is equivalent to the noise standard deviation. Besides voltage noise, the second major contributor to BER is timing uncertainty at the receiver. Like voltage noise, this uncertainty is a random process, and it is characterised by the jitter of the receiver clock as well as that of the transmitted signal. Both sources of jitter shift the sampling point away from its optimum, and have the effect

of reducing the voltage margin and degrading the BER. This effect is of particular concern as data rates increase, since jitter can become a substantial portion of a data period (also known as a unit interval, UI). As a result, timing margin can become a larger concern than voltage margin in high-speed links [19]. A helpful and common tool for visualizing the effects of noise and jitter on a link is the eye diagram, which is generated by superimposing many UIs of the data signal on top (Figure 2.2).

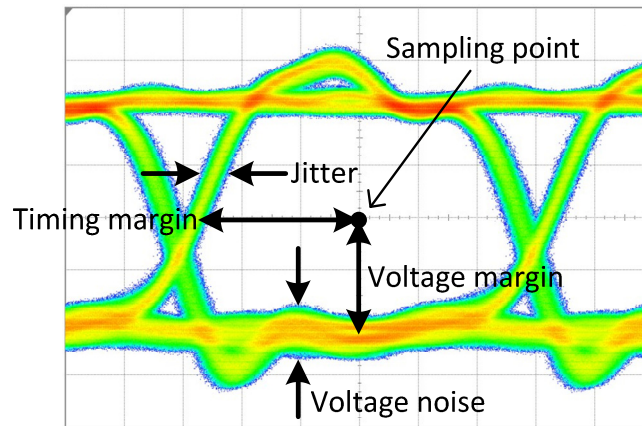


Figure 2.2: Eye diagram, showing sources of noise and timing and voltage margins.

The simple link structure described in Figure 2.1 forms the foundation for virtually all chip-to-chip communication in modern computing systems; indeed, for lower data rates, it can be used essentially unmodified. At higher (>1 Gb/s) data rates, however, a number of issues make this simple structure untenable, and designers have adopted a set of enhancements to deal with them.

2.1 Clocking

One of the challenges that arise at higher data rates is timing and synchronization. As the UI size, or bit time, decreases, the receiver has smaller and smaller timing margin and clocking naturally becomes more difficult. In order to provide a framework for discussion on this subject, it is helpful to outline several common clocking styles:

- **Synchronous:** from the Greek root meaning ‘same’. In a synchronous link, the transmitter and receiver clocks are assumed to have the same frequency and phase. This is generally only a tenable assumption at low data rates.
- **Mesochronous:** from the Greek root meaning ‘between’. In a mesochronous link, the transmitter and receiver clocks are assumed to have the same frequency, but may be out-of-phase. A popular sub-set of this category is the **source-synchronous** link, where the clock is generated at the transmitter and forwarded along with the data.
- **Plesiochronous:** from the Greek root meaning ‘near’ or ‘similar’. In a plesiochronous link, the transmitter and receiver clocks may have slight differences in frequency. The receiver is required to align its clock by extracting timing information from the incoming data stream.
- **Asynchronous:** an asynchronous link is not really clocked at all. Rather, it uses either control symbols inserted in the data stream itself or handshaking signals to convey timing information.

As the mesochronous/source-synchronous and plesiochronous styles are most frequently adopted for high-speed interconnect design, they shall be the focus of the discussion here.

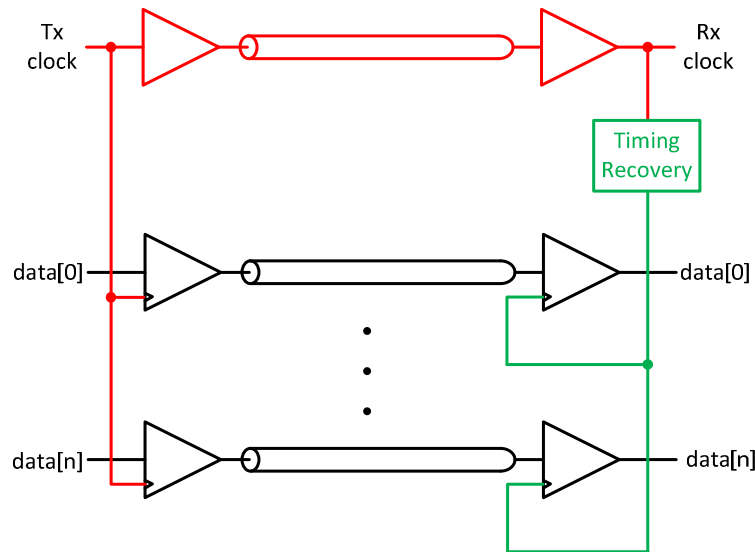


Figure 2.3: Source-synchronous link with shared timing recovery.

Since they require relatively straightforward timing recovery at the receiver (when compared to plesiochronous links), source-synchronous links are frequently used in computer systems, particularly where the link is composed of many data pins and the relative cost of adding a clock pin is small. Examples of source-synchronous links include memory interfaces such as DDR3 [20], and chip-to-chip interfaces such as HyperTransport [21] and QuickPath [22]. A typical block diagram is shown in Figure 2.3. With the clock forwarded along with the data, it will (ideally) experience similar amounts of skew and jitter, simplifying the task of synchronization at the receiver [23]; frequently, a single timing recovery block can be used across a bundle of many data pins that have the same origin. This assumption becomes untenable at higher data rates, however, where fabrication tolerances (e.g. of trace length) result in phase variation between data pins that is significant relative to a UI.

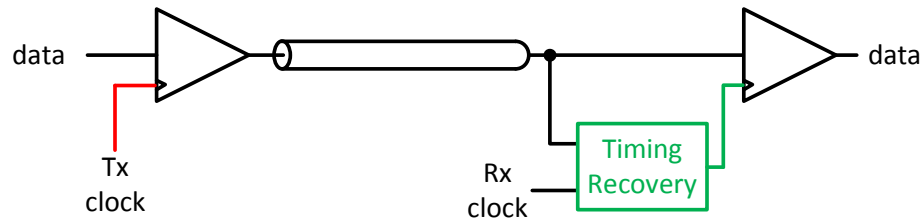


Figure 2.4: Plesiochronous link.

Where links are not as highly parallel and/or the distance to be spanned is longer, interconnect designers switch to plesiochronous schemes. This style of link does not require a forwarded clock, but complicates synchronization at the receiver by requiring it to extract timing information from the incoming data stream, and synchronize a local clock to it (Figure 2.4). The lower routing overhead makes plesiochronous links popular for communication between add-in cards and over server backplanes (e.g. PCI-Express [24]), which generally have to travel cover longer distances than the source-synchronous links described previously.

2.1.1 Sub-rate clocking

In a classical ‘full-rate’ link, the period of the clock is the same as the length of a UI and, for example, a 5 Gb/s link will operate with a 5 GHz clock. At multi-Gb/s data rates, however, the high-frequency clocks required for this approach consume large amounts of power and complicate

the process of timing recovery. As a result, designers use sub-rate clocking schemes. These are essentially multiplexing/demultiplexing schemes, where the clock operates at some integer fraction of the data rate and the data is transmitted and/or received using multiple phases of a clock period (Figure 2.5). Although it is, in principle, possible to generate as many phases of the clock as desired and lower the clock rate arbitrarily, practical concerns typically limit link implementations to half- and quarter-rates; in a half-rate link, the positive (0°) and negative (180°) edges of the clock can be used directly, and it is fairly straightforward to generate in-phase (0°) and quadrature (90°) clocks and their negations (180° and 270°) for quarter-rate systems. Half-rate systems are particularly popular (commercial examples include the DDR SDRAM, HyperTransport and QuickPath interfaces mentioned earlier), since clock generation is typically done differentially and the negated clock is essentially free. Another benefit of half-rate systems is that they match the bandwidth requirements of clock and data, which is particularly useful in source-synchronous links where clock and data are transmitted over similar channels.

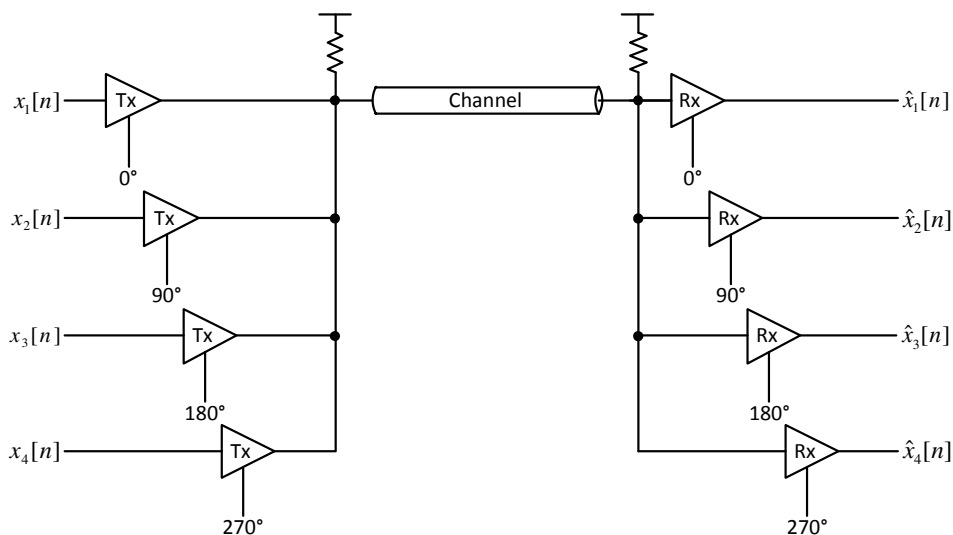


Figure 2.5: Quarter-rate link.

2.2 Clock and data recovery

As seen in the previous section, timing recovery is an important part of a high-speed link. In a mesochronous or source-synchronous link, the frequency of the incoming clock and data are

assumed to be the same, and the task of the timing or clock and data recovery (CDR) is to estimate and align their phases. This task gets considerably more challenging in a plesiochronous link, where the CDR must also compensate for frequency differences between clock and data.

Classical CDR techniques² can be split into two broad classifications: 2x oversampled and baud rate. A generalized 2x oversampled CDR is shown in Figure 2.6. The first phase of the clock is sent to the data slicer, and is placed near the middle of the eye. On its own, the data slicer provides no useable information about phase. Therefore, a second clock is used to slice the incoming signal at some phase offset from the data clock. The phase detector makes a comparison between the outputs of both slicers to determine the phase of the incoming data, and this information can in turn be filtered to generate a control signal for the clock generator that produces the two clock phases. This technique is appealing because the phase detector used takes inputs that have already be sliced into digital values, and is therefore relatively straightforward to implement in highly-scaled process technologies. Additionally, its operation is largely pattern-independent; system bandwidth is determined only by transition density, rather than the presence of particular bit patterns. The primary disadvantage is the requirement for the generation and distribution of two clock phases, which can increase the power consumption of such a CDR.

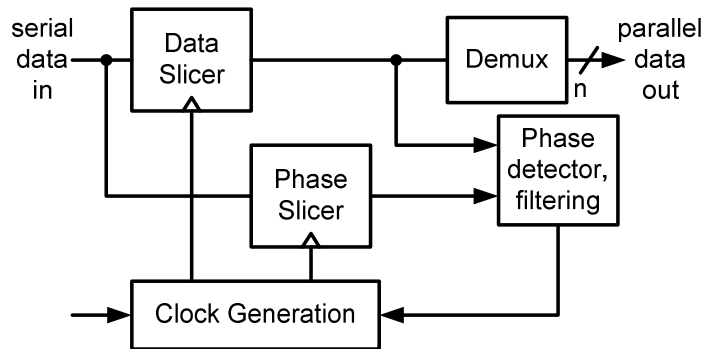


Figure 2.6: Generalized 2x oversampled CDR.

² For simplicity, the discussion in this section assumes full-rate clocking

A baud rate CDR (as the name suggests) uses sampler(s)³ operating on only a single phase of the clock, operating at the symbol, or baud, rate (Figure 2.7). Phase information is extracted by comparing characteristics (such as the voltage) of the present sampler output with those obtained previously. Requiring only a single clock phase gives the baud-rate approach a clear advantage over 2x oversampled CDRs; however, the analog processing required makes integration of such CDRs difficult⁴. Techniques to overcome this limitation are described below.

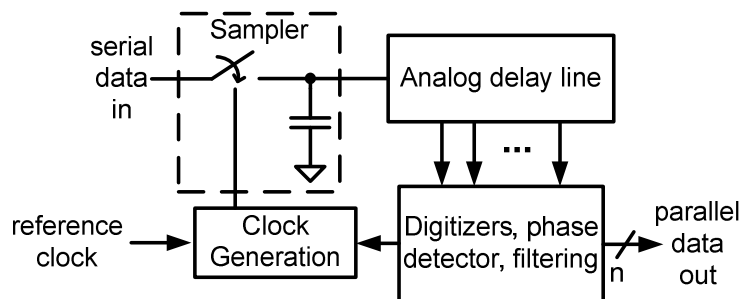


Figure 2.7: Generalized baud rate CDR

2.2.1 Phase detectors for 2x oversampled CDR

One classic example of a phase detector for use in a 2x oversampled CDR is the Hogge, or linear, phase detector, first described in 1985 [25]. The Hogge phase detector (Figure 2.8) operates by detecting transitions in the incoming data and generating pulses whose width is proportional to the phase difference between clock and data. Since the average of this output (Y in Figure 2.8) is dependent on the data transition density, a reference half-clock pulse (Z) is generated at every data transition and subtracted from this phase difference. The result can be integrated and used

³ A note about terminology: a ‘sampler’ converts a continuous-time input into a discrete-time output, but does not make any decision about what bit this output represents (hence it is still analog, or continuous-value). Conversely, a ‘ slicer ’ combines the functions of sampler and decision element, and converts a continuous-time input into a digital value that is both discrete-time and discrete-value. Therefore, a 2x oversampled CDR uses slicers, since it operates on digitized values, while a classical baud rate CDR uses samplers, since it requires analog values.

⁴ An exception to this is links which use ADC-based receivers. Though a full discussion of this type of receiver is outside the scope of this document, it is worth noting that, since the ADC provides (quantized) voltage information and operates at the baud rate, baud rate CDR is a natural fit.

to control a clock generator, which will lock half a UI away from the edge of the eye, as long as the clock duty cycle is 50%. Note that this sampling point is optimal so long as the eye is symmetric, which is not always the case.

The Hogge phase detector has a number of drawbacks that limit its utility in high-speed digital systems. For instance, its output pulse width is proportional to the residual phase error, so good resolution requires narrow pulses. Since these pulses must naturally be a small fraction of a UI, producing them requires fast, power-hungry XOR gates. This resolution limit is compounded by the static phase offset due to the non-zero (and likely mismatched) clock-to-Q delays of the flip-flops. Hogge's discrete implementation corrected for this offset by introducing adjustable delay lines, but process, voltage and temperature (PVT) variations make this approach unsuitable for integrated designs. Finally, the linear output of the Hogge phase detector requires analog processing to control the clock generator, and such processing incurs excessive power and area penalties in deep sub-micron digital systems. As a result, most recent CDR implementations (e.g. [26], [27]), are based on an older phase detector first proposed by Alexander in 1975 [28], which avoids these limitations and generates non-linear output that can be sent directly to a charge pump or digital loop filter.

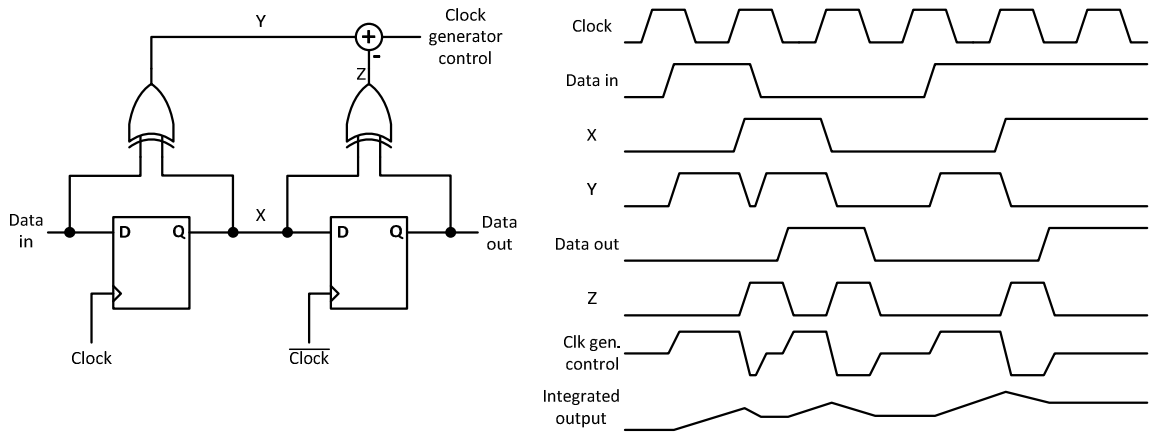


Figure 2.8: Hogge phase detector and timing diagram, showing integrated output gradually rising in response to phase difference between data and clock.

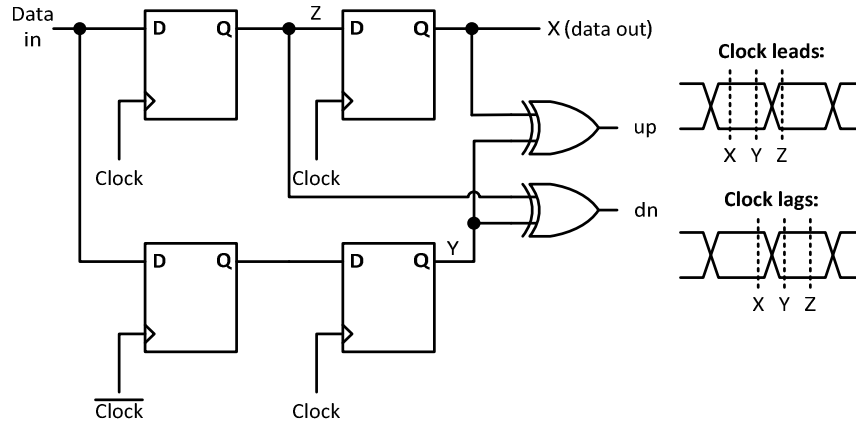


Figure 2.9: Alexander phase detector, showing sample locations when clock leads or lags.

X	Y	Z	up	dn
0	0	0	0	0
0	0	1	0	1
0	1	0	X	X
0	1	1	1	0
1	0	0	1	0
1	0	1	X	X
1	1	0	0	1
1	1	1	0	0

Table 2.1: Truth table for Alexander phase detector. ‘X’ states indicate don’t cares, since $X \neq Y$ cannot normally co-exist with $X = Z$.

The Alexander phase detector (Figure 2.9) operates by comparing samples taken on both edges of the clock. The basic principle of operation is to look for transitions in the incoming data ($X \neq Z$), then to check if $X = Y$, which indicates if the clock is leading or lagging the data. Based on this information, it generates ‘Up’ or ‘Down’ pulses to move the phase of the clock, an action known as ‘bang-bang’. Table 2.1 shows the truth table for this operation. The two don’t care states allow a degree of logic simplification; alternatively, they can be used to detect failure states [29]. Due to its non-linear nature (its output only has three states), a CDR based on an Alexander phase detector will eventually settle in a limit cycle about the lock point, creating

higher output jitter than a comparable Hogge-based system. The size of the limit cycle is related to the gain and delay through the CDR feedback loop; larger gain and longer delays produce larger limit cycles and more jitter. Note that, as in the Hogge phase detector, the Alexander phase detector is sensitive to clock duty cycle and makes the assumption that the eye is symmetric, placing the clock half a UI away from the eye edge.

2.2.2 Phase detectors for baud rate CDR

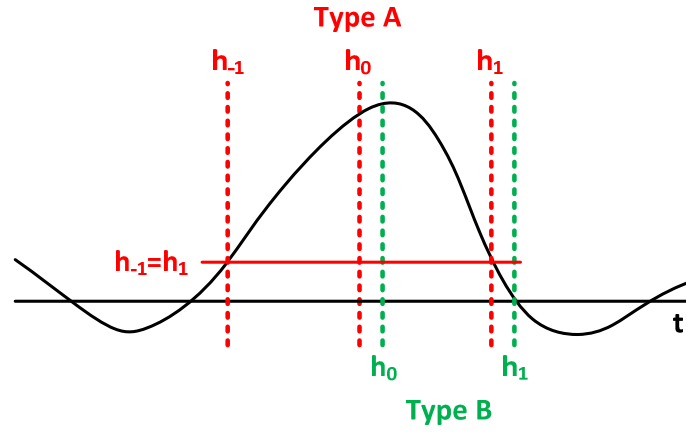


Figure 2.10: Example impulse response (adapted from [30]), showing the sample points of converged Mueller-Müller CDR, comparing Type A (red) and Type B (green). Sample points are UI spaced.

The prototypical baud rate CDR was proposed by Mueller and Müller in 1976 [30]. It uses samples of the incoming data to estimate the impulse response of the channel, adjusting the data clock such that the impulse response of the preceding and succeeding sample points (Figure 2.10) have certain behaviour. The original paper proposed two variations, *Type A*, which forces $h_{-1} = h_1$, and *Type B*, which forces $h_1 = 0$. Comparing the two types in the presence of phase distortion, the authors noted that the Type A system worked best when the impulse response was close to symmetric, while the Type B system was more tolerant of asymmetries in the impulse response. However, the Type B estimation function is much more complex and does not lend itself well to implementation. As a result, the focus of current design has been on variations of Type A, implementing the function:

$$h_1 - h_{-1} = E[x_k a_{k-1} - x_{k-1} a_k] \quad (2.2)$$

and adjusting the clock phase such that it is minimized. A typical architecture is shown in Figure 2.11, where x_k is the voltage value of the k^{th} sample, and a_k is the digitized value of the k^{th} sample (+1 or -1 for binary data). The output of the phase detector is sent to the loop filter of a clock generation loop, which performs the low-pass filtering necessary to determine the expected value.

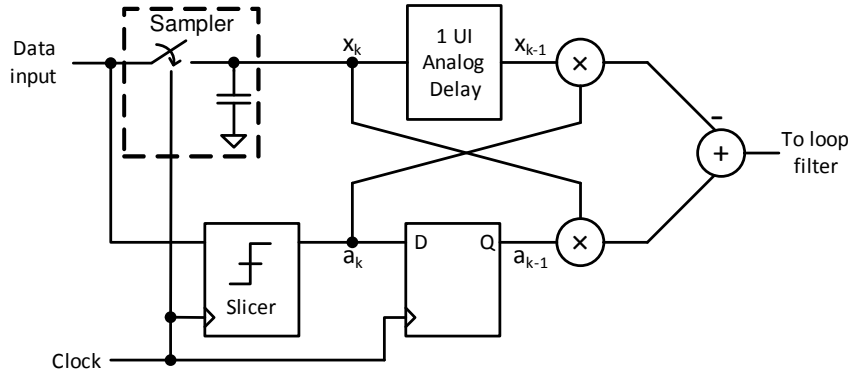


Figure 2.11: Architecture of Mueller-Müller Type A CDR.

Due to the requirement to delay and process the analog x_k values, implementations of the unmodified Type A system are impractical in highly-scaled CMOS. Realizations of this form of CDR instead rely on a sign-sign reduction of it [31], [32]. Extra slicers with positive and negative threshold offsets from the nominal decision point are used to produce error signals (ε_k ; $\varepsilon_k = +1$ when the input exceeds the boundaries defined by the thresholds, and $\varepsilon_k = -1$ when it is within the thresholds) that approximate the x_k values, so (2.2) becomes:

$$h_1 - h_{-1} = E[\varepsilon_k a_{k-1} - \varepsilon_{k-1} a_k] \quad (2.3)$$

Note that the extra samplers do not constitute a substantial hardware overhead, since their output can also be used to drive a SS-LMS adaptive equalizer (see Section 3.5, also [33]).

2.2.3 Clock generation

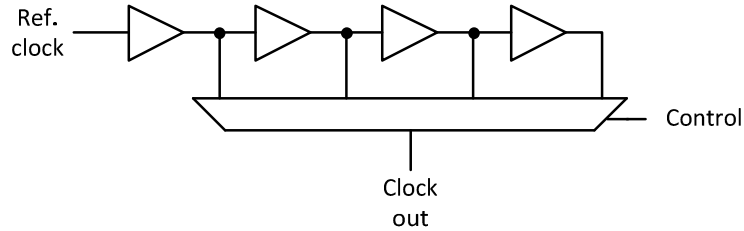


Figure 2.12: Open-loop delay line with 4 delay elements.

To complete the CDR feedback loop, the output of the phase detector has to be used to control some form of clock generator. Depending on the clocking scheme used, the generator can range from the very simple (open-loop delay line for source-synchronous/mesochronous links) to the complex (phase-locked or delay-locked loop for plesiochronous links with large frequency offsets).

The most basic clock generator for timing recovery is an open-loop delay line with variable-phase output (Figure 2.12). It is simply a series of delay elements that takes a reference clock as its input, with a multiplexer to select the appropriate delay element's output to feed to the samplers. The multiplexer can be controlled by the up/dn signals from a bang-bang phase detector. Note that this is not really a clock generator at all, but is more accurately described as a programmable phase shifter. Due to its open-loop nature, it places no guarantee on the exact amount of delay that it realizes; as a consequence, it can only correct for phase offsets as large as the length of the delay line. Any frequency offset between clock and data (as in a plesiochronous link) will eventually cause the delay line to run out of range and the CDR to break. Even in mesochronous links this approach faces some challenges. For example, it needs to be at least as long as the largest expected jitter transient, and the longer the delay line is, the more noise it accumulates and power it consumes. The resolution of the delay line is also limited to the fastest realizable delay element; optimistically, this is an inverter with fan-out of 1 (note that the load imposed by the output multiplexer and wiring ensures that this is, in fact, unachievable). Since the size of a bang-bang CDR's limit cycle is at least as large as its clock generator's smallest step size, this results in unacceptable output jitter.

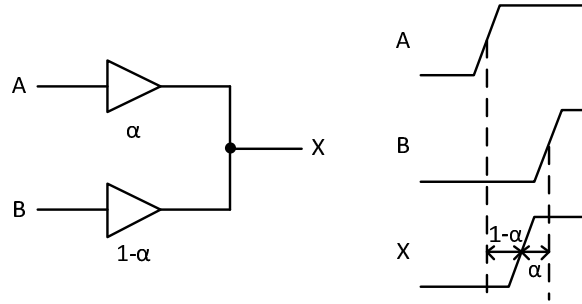


Figure 2.13: Phase interpolator, with weighted buffers ($\alpha < 1$). Output timing shown in ideal linear case.

The resolution of a delay line can be enhanced through the use of a phase interpolator (Figure 2.13), which takes two adjacent delay line outputs and produces a weighted average of their phases. In principle, there is no limit to the resolution that can be achieved using such a technique, but practical considerations (matching, jitter, power, and area) impose an effective upper bound on the resolution [34].

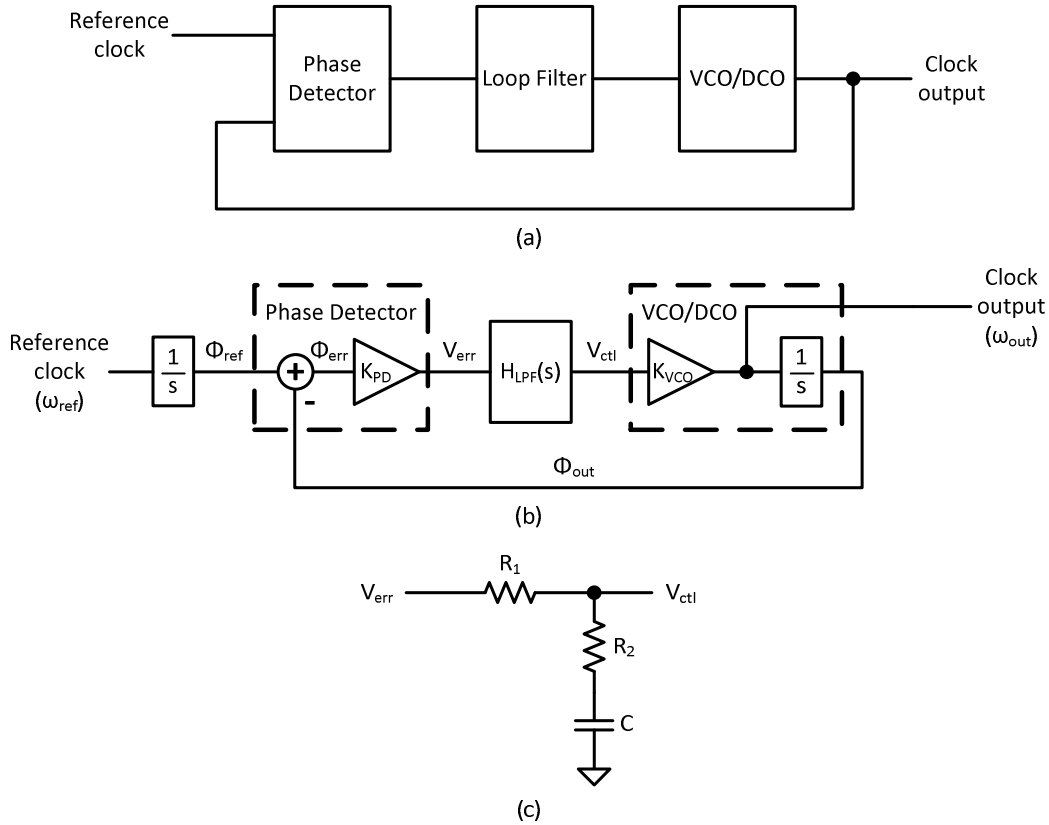


Figure 2.14: (a) PLL block diagram, (b) phase-domain model and (c) typical loop filter.

Increasing the resolution of the open-loop delay line through the use of a phase interpolator does not, of course, prevent it from running out of delay range. The solution to this problem instead requires a closed-loop approach, using a phase-locked loop (PLL) or delay-locked loop (DLL).

As its name suggests, the PLL works by synchronizing the phase of its internal oscillator to an external reference. In order to accomplish this, the PLL uses a phase detector to drive a loop filter, which generates a control signal for a voltage-controlled oscillator (VCO) or digitally-controlled oscillator (DCO) (Figure 2.14(a)). If the PLL is used purely for clock generation (e.g. to lock its oscillator to a clean reference clock), it can use a standard phase detector that expects regular transitions in the reference signal. If the PLL is used for CDR, the phase detector must be replaced by one sensitive to random data, such as the Hogge or Alexander types described in subsection 2.2.1.

The dynamics of the PLL can be analysed in the phase domain (Figure 2.14(b)). Note that the VCO or DCO generates a particular *frequency* based on its control input; since frequency is the derivative of phase, from the phase perspective the VCO/DCO looks like an integrator with some gain (K_{VCO}). In order to provide some means of controlling the PLL dynamics, the loop-filter is frequently a first-order low-pass filter (LPF) with one zero:

$$H_{LPF}(s) = \frac{1 + s/\omega_z}{1 + s/\omega_p} \quad (2.4)$$

where:

$$\omega_z = \frac{1}{R_2 C} \quad (2.5)$$

$$\omega_p = \frac{1}{(R_1 + R_2) \cdot C} \quad (2.6)$$

for the LPF in Figure 2.14(c).

As a result, the overall system looks like a classic second-order harmonic oscillator:

$$H(s) = \frac{\Phi_{out}(s)}{\Phi_{in}(s)} = \frac{K_{PD}K_{VCO}\omega_p(1 + s/\omega_z)}{s^2 + \omega_p \left(1 + \frac{K_{PD}K_{VCO}}{\omega_z}\right)s + K_{PD}K_{VCO}\omega_p} \quad (2.7)$$

Equating the denominator of $H(s)$ with the canonical form yields:

$$s^2 + 2\zeta\omega_n s + \omega_n^2 = s^2 + \omega_p \left(1 + \frac{K_{PD}K_{VCO}}{\omega_z}\right)s + K_{PD}K_{VCO}\omega_p \quad (2.8)$$

$$\Rightarrow \omega_n = \sqrt{K_{PD}K_{VCO}\omega_P} \text{ and } \zeta = \frac{1}{2}\left(\frac{\omega_P}{\omega_n} + \frac{\omega_n}{\omega_Z}\right)$$

Since the PLL is a second-order system, stability is an important concern, which complicates the design process considerably. Additionally, the phase-to-frequency conversion in the VCO/DCO results in phase error accumulation during noise or input transients [35]. Nevertheless, their phase noise filtering characteristics, inherent clock generation (without needing an external clock source) and straightforward adaptation as frequency multipliers means that PLLs continue to be a popular choice for CDRs.

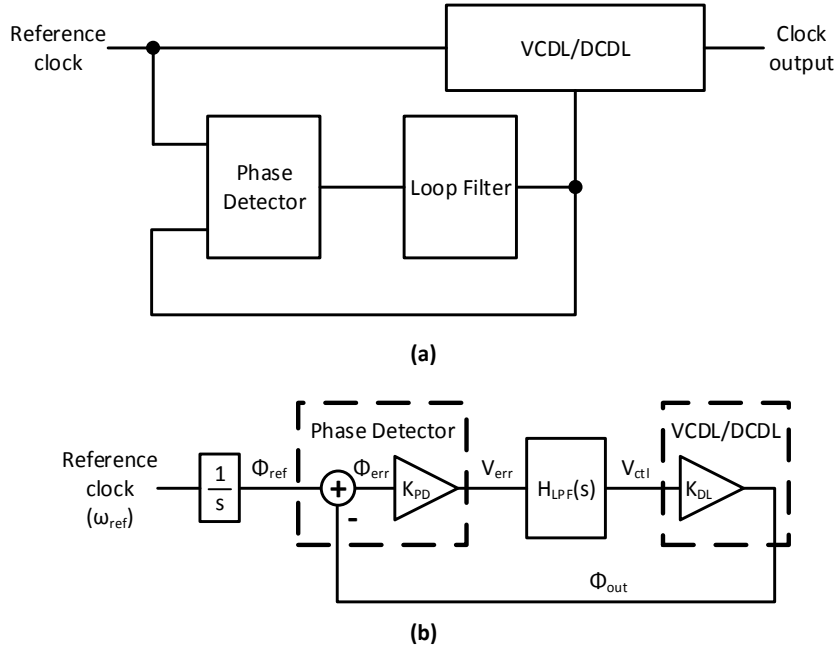


Figure 2.15: (a) DLL block diagram and (b) phase domain model.

The DLL avoids the stability and error accumulation concerns of the PLL by using a voltage- or digitally-controlled delay line (VCDL or DCDL) instead of the oscillator (Figure 2.15). Since the VCDL/DCDL converts its control signal directly into a phase shift, it is purely linear from a phase perspective. As a result, the overall loop dynamics are first-order and unconditionally stable. This simplifies the design process considerably, but means that the DLL translates the input phase noise to its output directly, without filtering. DLLs are also susceptible to false-locking states (where they spuriously lock to a delay that is some multiple of a UI, rather than

exactly 1 UI) and duty-cycle distortion, but these issues can be corrected relatively straightforwardly [35].

A larger problem with the classic, single-loop DLL is its inability to realize an infinite delay range. Just as with an open-loop delay line, the VCDL/DCDL will eventually run out of phase shift capability, causing the DLL-based CDR to break. The dual-loop DLL [36] solves this problem by introducing a second, peripheral, loop to perform phase alignment. The core DLL loop locks its delay at 180° , using inversion to achieve a full 360° delay range. The peripheral loop then selects outputs from this locked delay and interpolates between them in order to generate the final output phase. Since the core loop guarantees exactly 1 UI of phase shift, the peripheral loop can safely ‘wrap around’ the delay line without introducing any phase error, thus achieving an infinite delay range. It is also possible to base the dual-loop structure on a PLL acting as a multiplier for a lower-frequency crystal reference [26].

2.3 Equalization

At multi-Gb/s data rates, non-idealities in the communications channel ‘smear’ the sharp, well-defined transmitted symbols onto each other, creating inter-symbol interference (ISI). In many cases this effect is large enough to close the eye completely, making communication impossible without some form of compensation. ISI has two primary sources: dispersion due to frequency-dependent attenuation in the channel, and reflection due to impedance discontinuities. These effects can be visualized by plotting the pulse response of the channel (Figure 2.16), and the corresponding frequency response (Figure 2.17). Note that the ‘main cursor’ refers to the current bit, while the pre- and post-cursors are the ISI contributions of the current bit to sampling points before and after it, respectively.

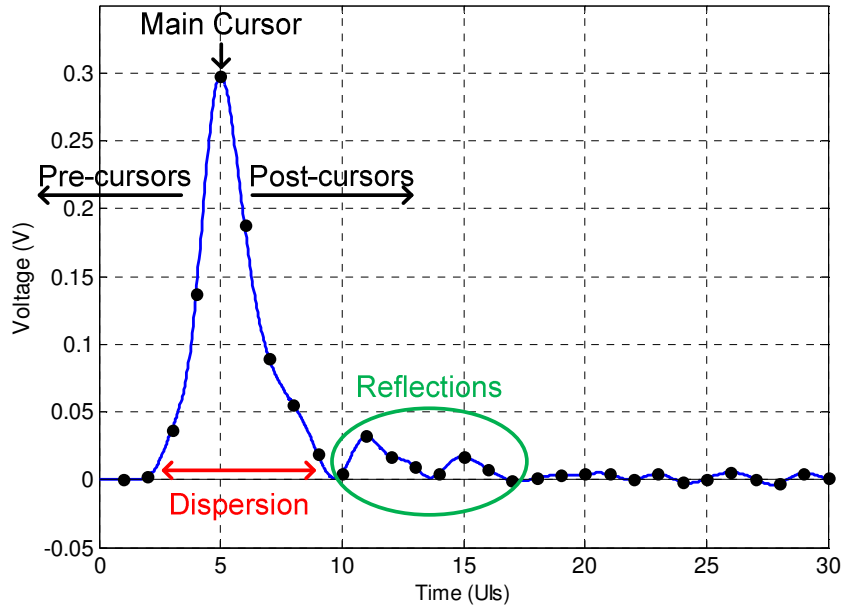


Figure 2.16: Pulse response of a legacy 16" server backplane at 12 Gb/s, showing sample points and the effects of dispersion and reflection.

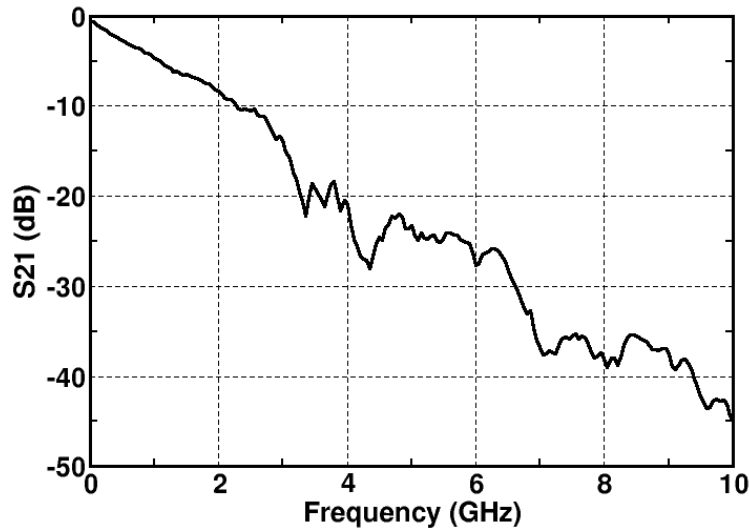


Figure 2.17: Frequency response of legacy 16" server backplane [37].

The frequency-dependent attenuation that causes dispersion has two main contributors: dielectric loss and skin effect. For reasons of cost and compatibility, chip-to-chip channels in server backplanes, desktop motherboards and add-in cards are frequently built from legacy

materials such as FR-4. These materials have poor dielectric loss properties (i.e. high loss tangents), which result in strong frequency-dependent attenuation (α_D):

$$\alpha_D = \frac{\pi f \sqrt{\epsilon_r} \tan \delta}{c} \quad (2.9)$$

where f is the frequency, ϵ_r is the relative dielectric permittivity, $\tan \delta$ is the loss tangent and c is the speed of light [18]. The attenuation due to dielectric loss is compounded by the skin effect; at high frequencies, the current through a conductor flows primarily along its surface. The severity of this effect is characterised by the skin depth, which measures the depth underneath the conductor's surface at which the current density has reduced to $1/e$ its value at the surface of the conductor:

$$d = \sqrt{\frac{\rho}{\pi f \mu}} \quad (2.10)$$

where ρ is the resistivity of the conductor and $\mu = \mu_0 \mu_r$ is the absolute permeability of the conductor [18]. The resistance-per-unit-length of a microstrip conductor can be written in terms of the skin depth:

$$R_s = \frac{\rho}{2wd} = \frac{\sqrt{\rho \pi f \mu}}{2w} \quad (2.11)$$

where w is the width of the conductor, and the effect of the sidewalls has been assumed negligible (since the height of a microstrip conductor, h , is typically much less than w). The resistance can be converted into an attenuation factor:

$$\alpha_s = \frac{R_s}{2Z_0} = \frac{\sqrt{\rho \pi f \mu}}{4Z_0 w} \quad (2.12)$$

where Z_0 is the characteristic impedance of the line. Notably, the attenuation factor due to skin effect is proportional to the square root of frequency, so it tends to have a smaller effect than dielectric loss at high frequency. As a result, choosing a dielectric with good loss characteristics can ease high-frequency link design considerably. However, economics and the desire to maintain compatibility with legacy backplanes frequently force designers to build transceivers that can work over older dielectric materials, which are very lossy at high frequency.

Understanding the second ISI contributor, reflection, requires some consideration of the geometry of the chip-to-chip channel. A typical server backplane (similar to that used to generate the pulse response in Figure 2.16) is presented in Figure 2.18. Impedance discontinuities occur

when the signal propagates through a changing electrical environment – e.g. at the connector between boards, or at vias. These impedance discontinuities cause part of the incident signal to reflect back on itself, and multiple discontinuities reflect pulses back-and-forth, creating the ringing behaviour in Figure 2.16.

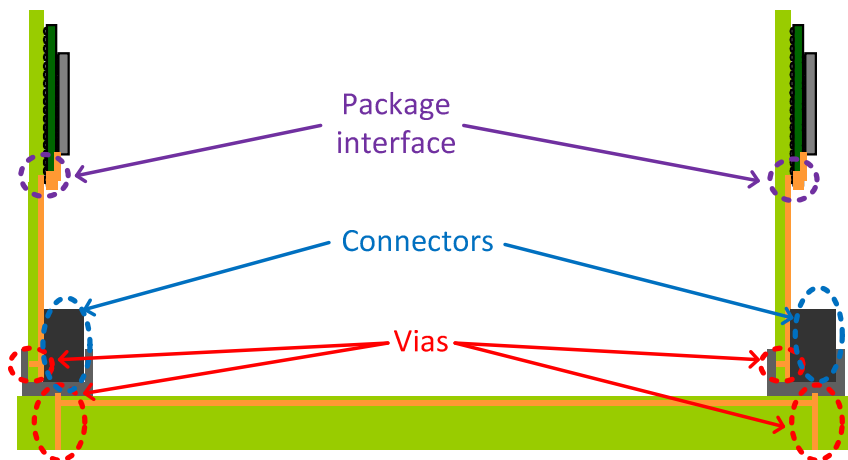


Figure 2.18: Typical server backplane, with major sources of reflections marked.

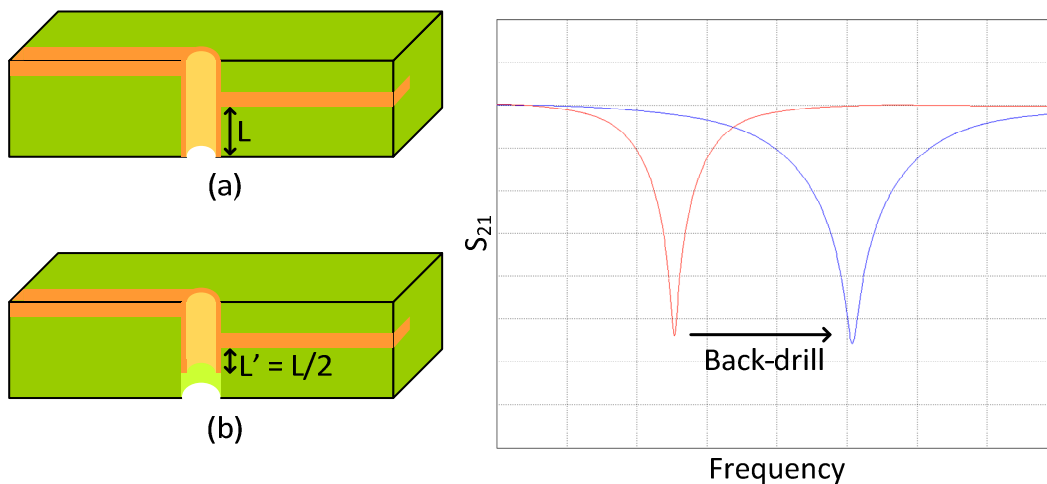


Figure 2.19: Via (a) before and (b) after back-drilling, showing effect on S_{21} (of the stub only).

Vias provide good examples of how reflections are generated, and what can be done to reduce or eliminate them entirely. A typical PCB via is manufactured by drilling a hole through the depth of the PCB, then plating it with metal. As a result, if it connects anything other than the outermost layers of the board, it will have a dangling tail, known as a stub. From a RF

perspective, this stub can be thought of as an open-circuit quarter-wavelength stub filter, which has a notch response at the frequency:

$$f_{notch} = \frac{c/\sqrt{\epsilon_r}}{4L} \quad (2.13)$$

where L is the length of the stub. The severity of the reflections can be reduced by back-drilling the stub, reducing L and pushing f_{notch} out to a higher frequency. Figure 2.19 shows an example where the back-drill reduces stub length by half, doubling f_{notch} . This helps in two ways: obviously, if f_{notch} is high enough, it can be beyond the communications bandwidth. Even if it is not, the attenuation of the channel is greater at higher frequencies, so shorter stubs will produce reflections that are more quickly damped out.

In order to allow successful communication in the hostile channel environment, designers employ a variety of corrective techniques, collectively known as equalization. The basic idea is to use knowledge about the channel response to design an inverse matched filter that can be placed in series with the channel (at the transmitter and/or the receiver), thus cancelling the ISI introduced.

2.3.1 Transmitter pre-emphasis

At the transmitter, the inverse filter is classically known as the pre-emphasis filter, since it pre-empts channel attenuation by amplifying (emphasizing) the high-frequency components of the transmitted signal. While this filter can certainly be realized using classic continuous-time linear filter design techniques, the area overhead of passive filters and the power overhead of active filters precludes the use of this design style for high-rate digital interconnect. Instead, a finite-impulse response (FIR) discrete-time filter is used (Figure 2.20). Each symbol in the sequence to be transmitted is fed through a series of delay elements, the outputs of which are weighted by the equalizer co-efficients and summed to yield the transmitted voltage. Each input to the summer is called a ‘tap’. In the most basic implementation, the taps are placed 1 UI apart, so, in addition to the main tap, each one corresponds to a single pre- or post-cursor ISI component. The filter response is adjusted by changing the various tap weights (h_{-i} through h_j) until the ISI contribution of each symbol is minimized.

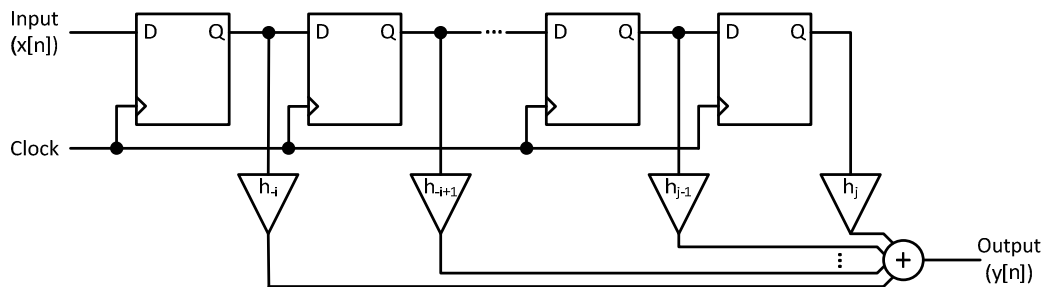


Figure 2.20: Discrete-time (FIR) transmitter pre-emphasis with i pre-cursor taps and j post-cursor taps.

Transmitter pre-emphasis faces several challenges, the most significant of which is the inability of the transmitter to sense the actual channel response. As a result, the tap weights must either be pre-set during fabrication, in which case they cannot adapt to channel variation over time, or a back-channel must be included in the system to transmit adaptation commands from the receiver back to the transmitter. The back-channel either occupies valuable board-level routing resources or requires complex duplexing schemes over existing wires (e.g. [38]). Additionally, the dynamic range available at the transmitter is limited, especially in deep sub-micron CMOS technology, where supply voltages have scaled below 1 V. In order to ensure that the output voltage never exceeds this limit, designs frequently attenuate the main cursor (a process sometimes called ‘de-emphasis’). The eye at the receiver is opened, therefore, at the expense of signal swing.

2.3.2 Receiver linear equalization

Linear equalization in the receiver is essentially the dual of transmitter pre-emphasis – a linear filter is applied to the received signal to invert the channel response. Since equalization in the receiver must be applied *before* the sampler (Figure 2.21), and therefore while the received signal is still in the continuous-time domain, this precludes the direct, efficient use of a FIR filter (unlike the case of transmitter pre-emphasis). As a result, the ability of receiver linear equalization to handle complex channel responses (e.g. reflections) is limited by the expense of implementing the higher-order analog filters required to invert these responses. Just as importantly, the incoming signal will have picked up noise (AWGN and crosstalk) in the channel.

Therefore, by applying an inverse filter that amplifies the high frequency, lower signal-to-noise ratio (SNR) components of the signal while suppressing the low frequency components, the receiver linear equalizer effectively acts as a noise amplifier and decreases the voltage and timing margins available at the sampler. Despite these disadvantages, receiver linear equalizers remain appealing because they are placed at the receiver and can therefore adapt to the channel without backchannel communication.

2.3.3 Decision-feedback equalization

The third major type of wireline equalizer is a decision-feedback equalizer (DFE). This type of equalizer attempts to overcome the limitations of receiver linear equalization by placing a filter in the feedback path, instead of the feedforward path. This filter can be thought of as a channel *emulation* filter, rather than a channel inversion filter, since its intent is to match the ISI created by the channel. The estimated bits are sent through the feedback filter to produce an ISI prediction, which is subtracted from the incoming signal to equalize it. Since the feedback filter accepts discrete-time input from the slicer, it is natural to implement the feedback filter as a FIR system (Figure 2.21).

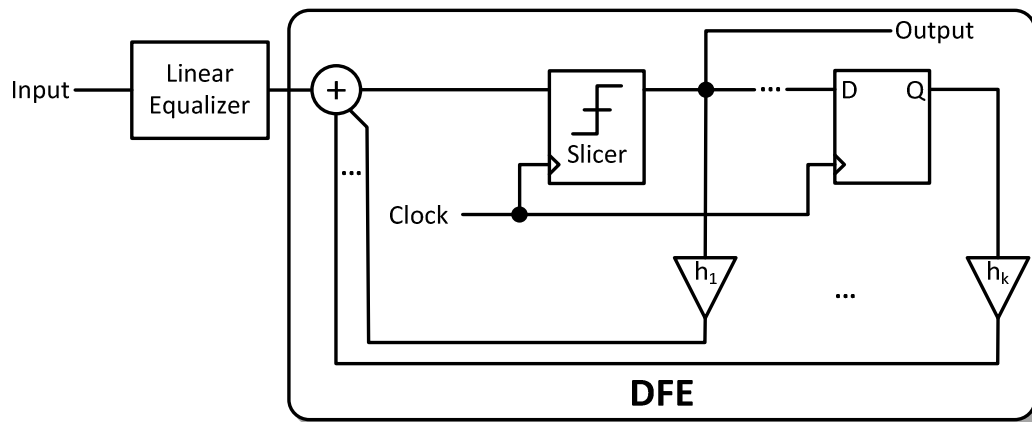


Figure 2.21: Receiver with linear equalizer and DFE.

The DFE acts by estimating and cancelling ISI, rather than inverting a channel response. As a result, it suffers less severe noise amplification than a linear equalizer [39]. However, to maintain causality, the DFE can only operate on post-cursor ISI. As a result, it is frequently used

in concert with linear equalizers (either in the transmitter or receiver), which cancel the shorter pre-cursor ISI [40]. Additionally, the DFE makes the assumption that the previous bits have been correctly estimated – if any error in estimation has been made, this can propagate through the DFE and affect subsequent bits. Wireline communication systems are typically designed to operate at very low BER (e.g. $<10^{-12}$ in HyperTransport [21]), so this tends not to be a significant impediment.

Although the discrete-time FIR implementation makes a DFE amenable to integration in digital systems, high-attenuation channels with long post-cursor ISI tails require large numbers of taps that make implementation prohibitively expensive. This problem can be mitigated through the use of transmitter pre-emphasis (the transmitter taps undergo the same dispersion as the main cursor, so a small number of transmitter taps can be sufficient to cancel the long-tail ISI) [40]; alternatively, there have recently been proposals to use continuous time feedback filters to emulate the channel response and reduce DFE tap count [41].

Timing can also be a challenge, since the first-tap feedback loop must be closed within a single UI. To achieve the performance required for multi-Gb/s links, the feedback path and summer are typically realized using current-steering/summing architectures, and the first feedback tap is evaluated speculatively (i.e. the outputs for both possible values of the feedback tap are pre-computed, and the correct one selected when the slicer output is available) [42]. Such current-based architectures consume large amounts of power, so switched-capacitor summers have been proposed as a more efficient alternative [43].

2.4 Summary

A broad range of techniques for timing recovery and equalization has been developed to facilitate high-speed chip-to-chip communication over a hostile channel environment. Due to the fact that transmitting and receiving data across a communications channel is fundamentally an analog process, these techniques become more challenging as CMOS technology scales to ever finer geometries and transistors are increasingly optimized for use as digital switches rather than analog amplifiers. For example, closed-loop clocking systems such as PLLs and DLLs require

components such as voltage regulators, charge pumps and loop filters, which rely on the matching and/or amplification characteristics of transistors to perform well; these properties are severely degraded in highly-scaled CMOS. At the same time, CMOS scaling allows an increase in parallel computing power, consequently increasing the demand for high-speed communication. Compounding this, in order to keep fabrication costs down, high-attenuation legacy materials such as FR-4 remain the substrates of choice for the fabrication of many high-speed interconnect channels (especially in the low-margin desktop and notebook markets), driving up the complexity and power consumption of equalization schemes. These factors have conspired to create the scaling disparity between communication bandwidth demand and power consumption noted in section 1.2. Addressing this disparity will require the innovative application of more-digital (or all-digital) techniques, as well as an increased reliance on sharing, parallelism and multi-functional, adaptive hardware.

Chapter 3: All-Digital Clock and Data Recovery

Source-synchronous links are among the most common types of interconnect used in chip-to-chip communication, since they experience a degree of correlated jitter and frequency offset tracking [23]. Consequently, at low data rates a single timing recovery block can be sufficient to align the incoming clock and data over multiple data pins with the same source and destination. However, as data rates scale, PVT variation at both the board- and chip-level can cause sufficient mismatch in delay between pins to require per-pin phase alignment [22]. Per-pin phase alignment also relaxes path-length matching requirements between traces, allowing greater signal densities over limited routing resources. Additionally, variability and noise in clock multiplication/distribution at both the transmitter and receiver blur the lines between source-synchronous and plesiochronous systems, by introducing frequency offsets and jitter transients substantial enough to create the need for full CDR with infinite delay range capability, albeit at bandwidths lower than a traditional plesiochronous link might require.

In order to achieve per-pin phase alignment, the power and area overheads of adding dedicated CDR circuitry to each data pin are prohibitive. This is particularly the case for traditional analog-based techniques such as PLLs or DLLs, which rely on VCOs, analog loop filters and charge-pumps. The design space for these components is quite different from that of the digital logic comprising the rest of the system; considerable area, power, design and manufacturing overheads are necessary to accommodate these differences. Recent work on CDR has, therefore, focused on more-digital techniques [44], [45], [46]. However, they still rely on analog components such as voltage regulators, resistors or varactors to control the core VCO. This chapter describes a true all-digital CDR scheme; except for the input slicers (StrongARM latches [47]), the system is implemented entirely in static CMOS logic. The algorithm itself is

tolerant of the effects of transistor mismatch, avoiding the need for complex calibration and compensation loops. Additionally, to make the system amenable to implementation in large digital systems, heavy use is made of standard cell blocks, automatic synthesis and place-and-route.

Eye-monitor-based CDRs provide an alternative to traditional 2x oversampled and baud rate approaches; instead of using an explicit phase detector, the data sampling hardware is replicated, and its output compared with the primary data path to determine BER and construct the eye opening. To date, eye-monitors have focused on two-dimensional (voltage and time) approaches for adaptive equalizers, but not for timing recovery [48], [49]. Others still rely on Alexander-type phase detectors and use the eye-monitor to fine-tune the data clock phase for improved error tolerance [50], [51]. Although [52] does incorporate timing recovery using only eye-monitor data, it requires external computer-based calibration. Other work has investigated the use of eye-monitors for off-line channel characterizations [53].

In contrast, the all-digital CDR described here [54], [55] focuses specifically on timing recovery, and conducts one-dimensional (time-only) eye-monitoring, avoiding the overhead of a variable-threshold input slicer. Section 3.5 describes how it is also possible to use a single-dimension eye-monitor for equalization adaptation. A key innovation in this algorithm is the use of ‘ping-pong’ clocks; the data and eye-monitor functions are swapped between clocks during updates of the data phase. This confers important advantages by insulating against mismatch between the phases of the clocks and allowing the realization of an infinite delay range using only a loosely-calibrated delay line, instead of a PLL or DLL. Additionally, the search technique is designed specifically for efficient on-chip implementation via hardware description language (HDL) synthesis.

Since an eye-monitoring CDR uses a BER-based (and therefore statistical) approach to finding the eye, it has relatively low bandwidth and is not appropriate for use in plesiochronous links, which require large (hundreds of ppm) frequency offset tolerance. However, it can be applied to mesochronous or source-synchronous links; as mentioned above, at high data rates, even historically small amounts of transmission path length mismatch or noise can result in multiple-UI jitter at the receiver and the need for infinite delay range-capable CDR.

This chapter also discusses the application of sharing techniques that allow a single set of CDR hardware to calibrate multiple data pins, taking advantage of the relatively low bandwidth requirements of source-synchronous links. The discussed algorithm is particularly well-suited for sharing, since the sampling and re-timing hardware required for the ping-pong clocks can be reassigned between data pins.

3.1 Ping-Pong CDR

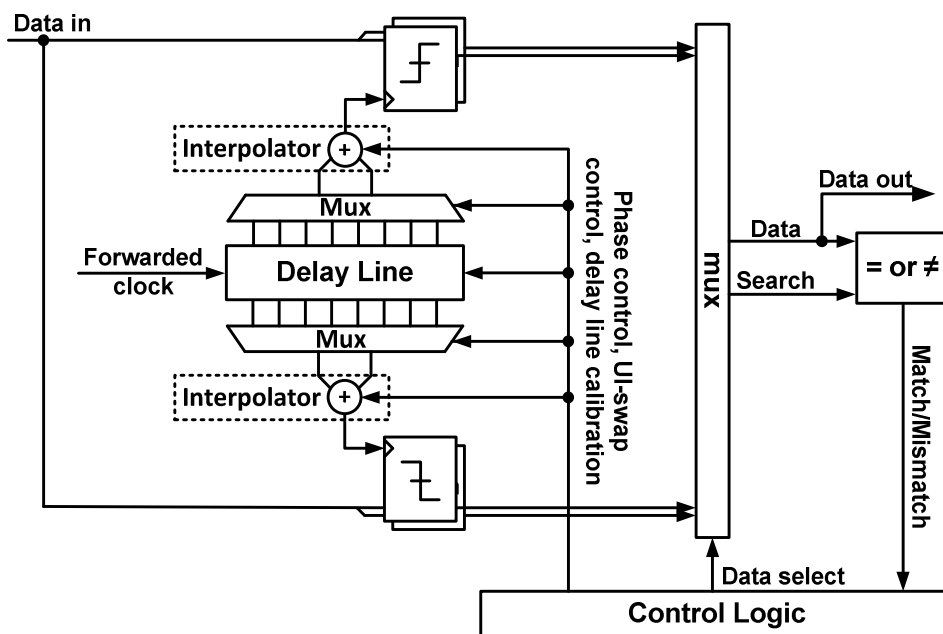


Figure 3.1: Single-pin ping-pong CDR architecture overview.

Section 2.2.1 describes random-data phase detectors traditionally used in CDR. These 2x oversampled approaches use two clocks to sample the incoming signal - the first is used to sample the data, while the second is used to sample the edge of the eye. In a half-rate system, this means that the edge clock is fixed at 90° phase offset from the data clock. In eye-monitor-based CDR, the second clock's phase is decoupled from the first and allowed to move independently. The ping-pong CDR uses similarly decoupled clocks, with both clocks free to move at discrete intervals ('phase positions') within a 2 UI delay. One clock is placed in the middle of the eye to recover data, and is known as the 'data clock'. The other is swept across the 2 UI delay, and is

known as the ‘search clock’. BER data is collected by comparing the samples produced by these clocks. This BER data is used to reconstruct an approximate eye diagram and determine the best phase position for data recovery. The search clock is placed at this phase position, the search and data functions are traded between clocks (the ‘ping-pong’), and the algorithm repeats.

A delay line is used to generate the two clocks from the source synchronous link’s forwarded clock (Figure 3.1). This delay line is slowly and digitally calibrated to achieve approximately 2 UI delay (as explained below, exact calibration is unnecessary). Adjacent output phases of the delay line are selected and interpolated independently for each clock. A multiplexer following the input slicers routes their outputs to either ‘search’ or ‘data’. Synthesized digital logic aggregates this data to determine the location of the eye opening and controls the movement and swapping of the clocks.

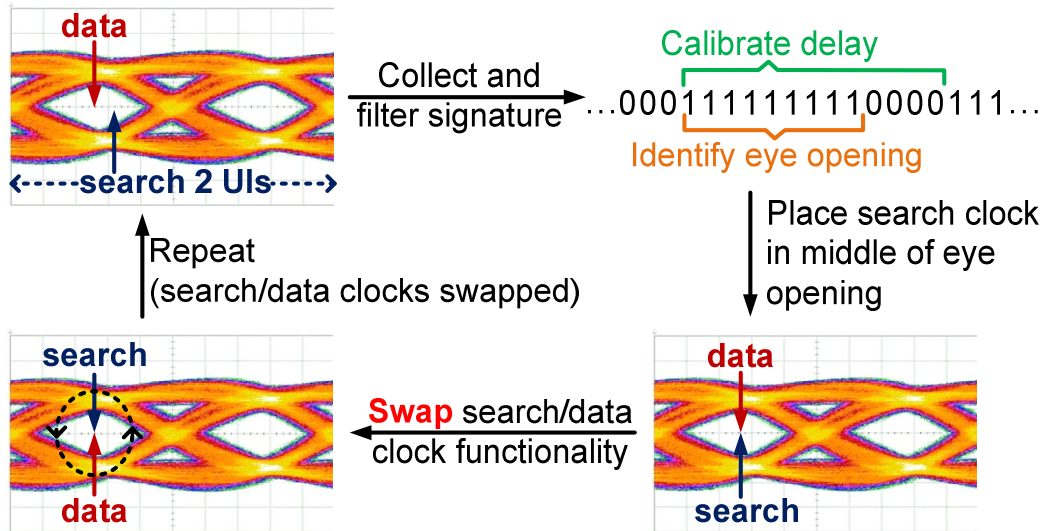


Figure 3.2: Ping-pong CDR algorithm.

The samples generated by the search clock are compared with those produced by the data clock. Where these samples match, the eye is open. Conversely, a mismatch between these samples indicates that the eye is closed. As the search clock is swept through the 2 UI delay line, match/mismatch information is collected at each phase position. The collated information can be thought of as a binary reduction of an eye diagram (Figure 3.2). Transitions between match and mismatch correspond to the edge of the eye, so the control logic can use these transitions to place

the search clock at the mean of the detected eye edges, maximizing timing margin. Note that, unlike in an Alexander or Hogge-based CDR, no assumption is made that the optimum sampling point is half a UI away from the eye edge. The functionality of the search and data clocks is then traded and the algorithm repeats. This ping-pong between search and data clocks allows the CDR to realize an infinite delay range. When the eye opening begins to drift off the extent of the delay line, the CDR can place the search clock in the middle of the following or preceding eye opening and invert it *before* initiating the ping-pong between search and data functionality. In a half-rate architecture, inverting the search clock before the ping-pong allows the CDR to skip backward (or forward) a UI without introducing errors such as added or dropped bits in the data. Figure 3.3 shows an example of a UI swap: in Step 1, the search process has just finished and the control logic has determined that a UI swap is required. In Step 2, the even (rising) edge of the search clock is placed in the previous UI (marked ‘3’). To line up the even and odd edges of the clock, the search clock is inverted in Step 3. Once this inversion is complete, the clock functions can be interchanged without added/dropped bits (Step 4).

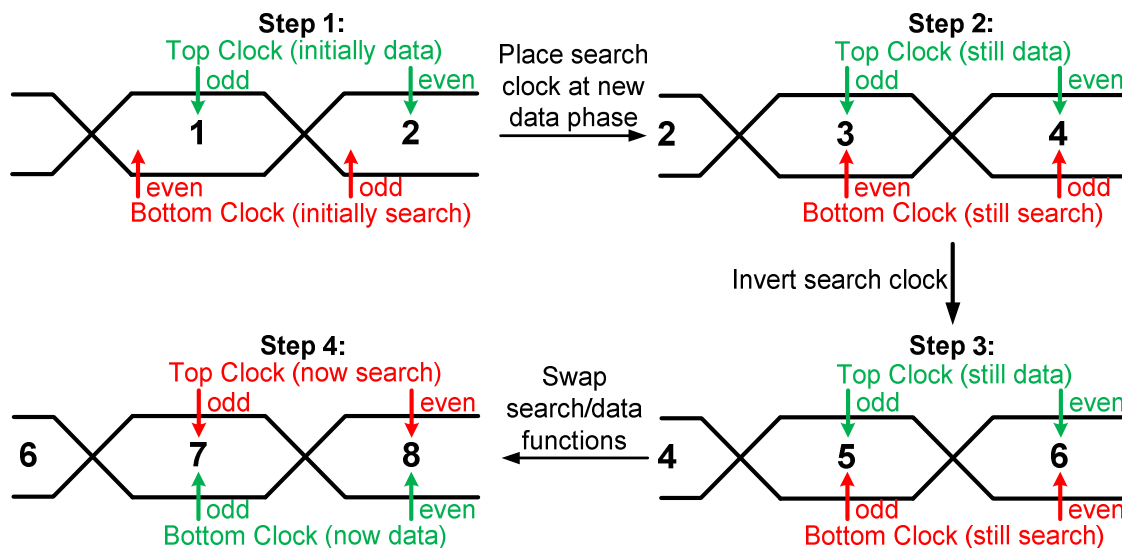


Figure 3.3: Steps in a UI swap process. Even (rising) and odd (falling) edges of the clocks marked.

Match/mismatch data is generated by sweeping the search clock's phase, so the eye information collected is based on the actual phase shift introduced by the search clock's own

phase generation and distribution path. Since the search clock uses this information to determine its sampling point when it becomes the data clock, accurate data clock placement is not dependent on matching with any other clock's phase generation and distribution. This advantage of the ping-pong CDR is particularly important in a multi-pin environment, where many clocks are required and matching between their paths becomes prohibitively difficult.

3.1.1 Search algorithm

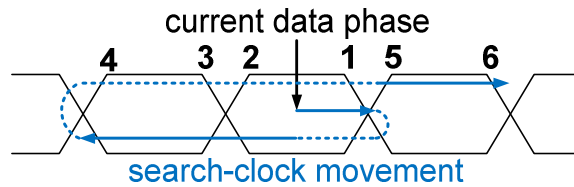


Figure 3.4: Search procedure, showing movement of search clock and numbered eye edges.

The CDR spends the vast majority of its operating time sweeping the search clock through the delay line and collecting BER statistics to find the eye. It is therefore vitally important to optimize this search process, in order to maximize CDR bandwidth. In normal operation, it is unnecessary to search the entire 2 UI delay in order to find the eye and update the data clock. Instead, operation is hastened by stopping the search once two edges bounding a single eye opening are detected (marked 1 and 2 in Figure 3.4). The search clock movement described in Figure 3.4 minimizes the time to find these bounding eye edges. A more complete search is only conducted when the eye opening begins to drift off the extent of the delay line, or if a delay line calibration is requested. For example, Figure 3.5 presents a case where the phase offset between clock and data is gradually increasing due to a frequency offset or large jitter transient. As data drifts to the right, the CDR tracks it and updates the data phase accordingly. When the data drifts far enough, one edge of the current eye moves off the end of the delay line and cannot be found. The CDR then searches for eye edges 3 and 4 to acquire the previous eye opening. It places the search clock in the middle of this eye opening, inverts the search clock and does the ping-pong between search and data functionality, completing the data clock update.

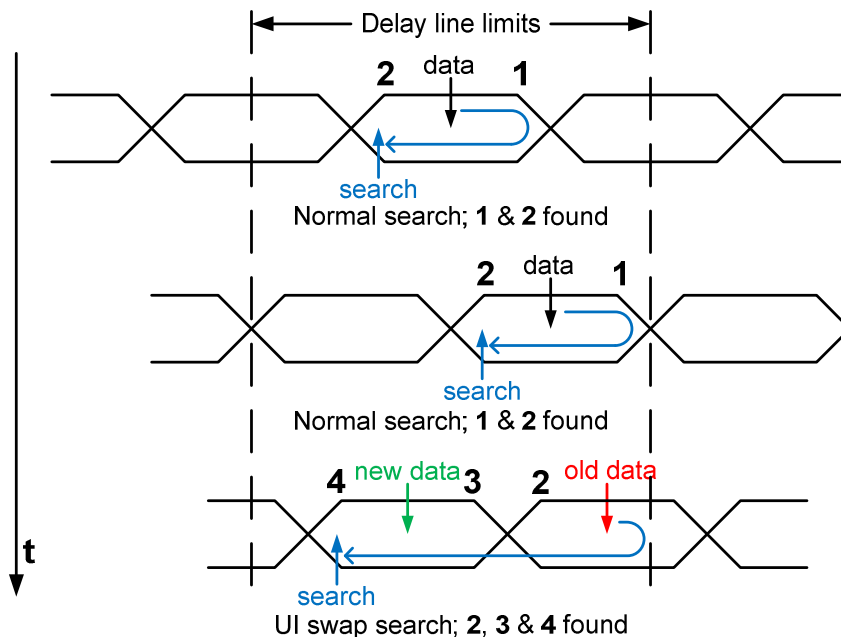


Figure 3.5: CDR operation with eye drifting until UI swap is required. Relevant eye edges are marked and numbered.

A longer search is also required to calibrate the delay line. In this case, the control logic seeks to establish the length of 1 UI in terms of phase positions, then adjust the length of the delay line such that 1 UI of delay occupies half the available phase positions (hence the overall delay will be 2 UI). This information is obtained by extending the search one further eye edge – to 3, or 5 if 3 is not found. The distance between eye edges 3 and 1 or 2 and 5 corresponds to the length of 1 UI. Exact calibration of the delay line is unnecessary, since the algorithm will never need to search through a full 2 UI; in the worst case, it only needs sufficient range to find the preceding or following eye opening in preparation for a UI swap. This calibration process can therefore proceed loosely and slowly, so it has limited impact on the overall bandwidth of the CDR.

3.1.2 Search filtering

Since eye-edge detection (hence, data phase placement) is reliant on the match/mismatch information collected, it is important that this be done as consistently and accurately as possible. The incoming signal will include data-dependent phenomena such as ISI and transient events such as noise spikes, which may cause spurious match/mismatch decisions and corrupt the

detection of the eye opening. These effects are suppressed through the use of a mismatch counter, which acts as a pre-filtering averager, and an AND/OR filter (Figure 3.6).

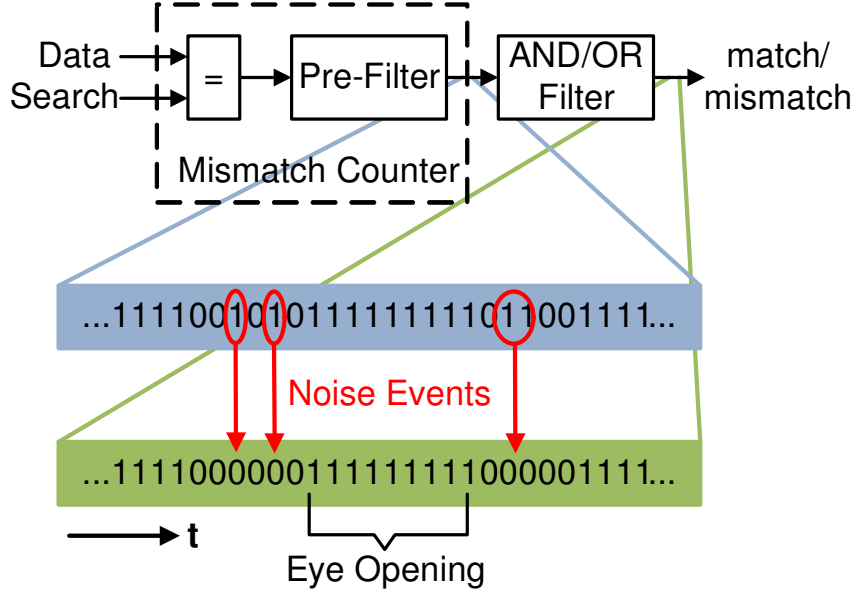


Figure 3.6: Eye information collection via mismatch counter and AND/OR filter.

The mismatch counter observes the incoming data stream; when a transition occurs, it makes a comparison between the corresponding search and data samples. Conducting averaging over n such transitions, the probability that at least one of the collected comparisons is conducted on dissimilar search and data samples (therefore, that phase position i will be declared a mismatch) is given by a geometric distribution:

$$P(\text{mismatch}) \approx 1 - (1 - BER_i)^{n/\rho} \quad (3.1)$$

where BER_i is the bit-error rate at the i^{th} phase position and ρ is the average transition density. Ideally, the search should transition from generating matches to mismatches at a consistent phase position relative to the eye opening, tracking it if it moves. Consider the case where the search is moving downwards from the data phase (i.e. from higher to lower phase position) and from generating matches to generating mismatches. In this case, the probability that phase position i will be the first mismatch generated (therefore, the location of the detected eye edge) can be defined recursively:

$$P(\text{firstMismatch}_i) \approx \left(1 - \sum_{j=i+1}^d P(\text{firstMismatch}_j)\right) P(\text{mismatch}_i) \quad (3.2)$$

for $i < d$, where d is the current data phase position, and it is assumed that $P(\text{firstMismatch}_d) = 0$. The most consistent results are produced when the distribution generated by (3.2) is tightest – ideally, concentrated on a single phase position. This suggests that the sides of the bathtub curve produced from (3.1) should be as steep as possible; this can be achieved by making n large. However, large values of n result in long averaging times at each phase position, reducing the bandwidth of the CDR.

A more reasonable approach is to semi-dynamically size n , collecting more samples only in ambiguous cases. To this end, a new averaging period, n_{base} , can be defined. If the search and data clocks produce two or more dissimilar samples in the first n_{base} transitions, the mismatch counter will immediately declare the phase position a mismatch. An ambiguity occurs when there is only one discrepancy between search and data samples in the first n_{base} transitions. In this case, the mismatch counter will collect samples for a further n_{base} transitions. If another discrepancy between search and data samples occurs, a mismatch will be declared. Conversely, if no discrepancy occurs, a match will be declared. With this ability to repeat the search, the probability of a mismatch declaration at phase position i becomes:

$$\begin{aligned} P(\text{mismatchWithRepeats}_i) &= P(\text{outrightMismatch}_i) + P(\text{repeatMismatch}_i) \end{aligned} \quad (3.3)$$

An outright mismatch occurs if there are two or more discrepancies in the first n_{base} transitions:

$$P(\text{outrightMismatch}_i) = \sum_{j=0}^{n_{base}/\rho} \binom{1+j}{j} BER_i^2 \times (1 - BER_i)^j \quad (3.4)$$

where BER_i is the bit error rate at phase position i . A mismatch-on-repeat occurs if the mismatch counter waits a further n_{base} transitions, and one or more discrepancies occurred during this extra collection period:

$$\begin{aligned} P(\text{repeatMismatch}_i) &= (P(\text{mismatch}_i) - P(\text{outrightMismatch}_i)) \\ &\times P(\text{mismatch}_i) \end{aligned} \quad (3.5)$$

$P(\text{mismatchWithRepeats}_i)$ can be substituted in place of $P(\text{mismatch}_i)$ in (3.2) to obtain the distribution of the first mismatch using the modified method. Figure 3.7 presents the probabilities

of mismatch declaration (using (3.1) and (3.5)) for $n = n_{base} = 32$. The slope of the bathtub with repeats is steeper than that without repeats, suggesting better performance. This result is made clearer in Figure 3.8, which charts the probability of match-to-mismatch transition detection with and without repeats.

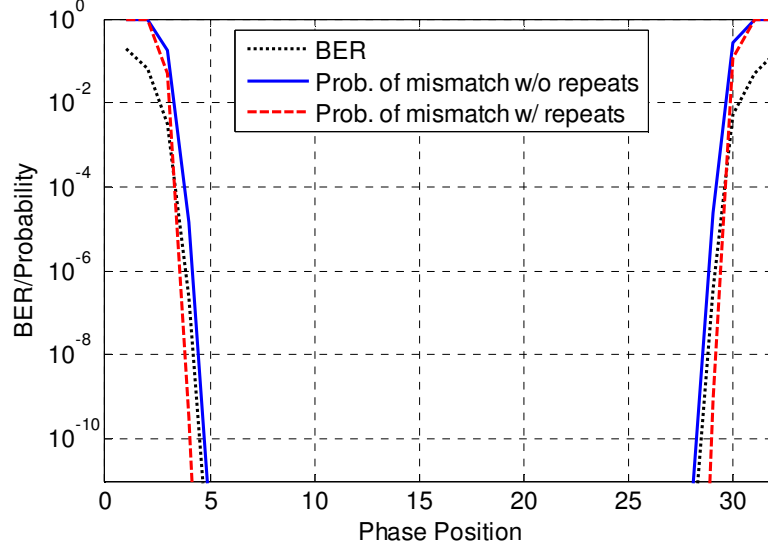


Figure 3.7: Typical BER bathtub, and the probability of mismatch declaration at each phase position with ($n_{base} = 32$) and without ($n = 32$) repeated averaging⁵.

⁵ The BER bathtub shown here is used in all subsequent figures in this sub-section, and for calculating of the optimum value of n_{base} . Different BER bathtubs will yield different results. The one used here is based on measurements of the actual channel used to test the ping-pong CDR.

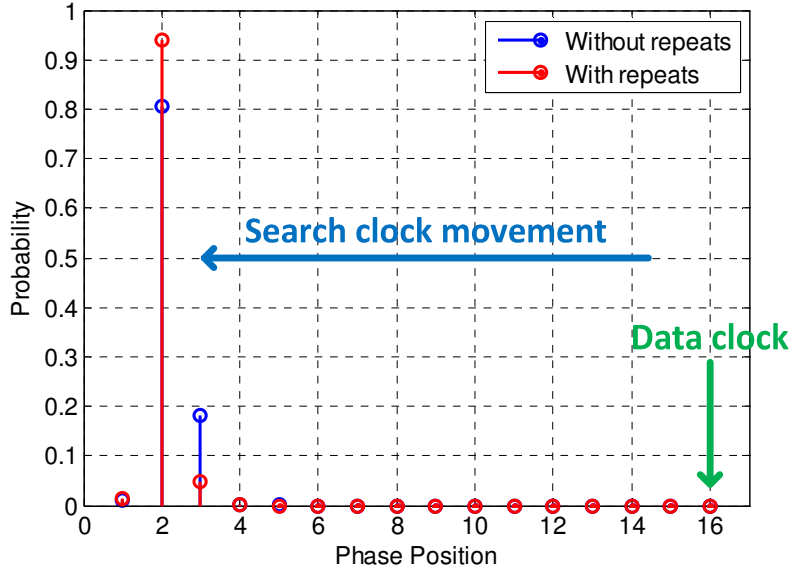


Figure 3.8: Probability of match-to-mismatch transition detection, $n = n_{base} = 32$.

For consistency of eye-edge detection, this probability should be as concentrated on one phase position as possible. A measure of this concentration is the value of the highest probability in the distribution such as Figure 3.8. Plotting this for reasonable values of n_{base} and n (Figure 3.9) shows that larger values of n initially provide more consistent detection, as might be expected. However, if n gets too large, lower BER regions of the eye become more likely to produce a mismatch, spreading the probability of match/mismatch transition detection between two or more phase positions and reducing consistency. Figure 3.10 shows the $n = n_{base} = 128$ case, where the probability of match/mismatch transition detection has spread between phase positions 2 and 3. Figure 3.9 suggests an optimum of $n_{base} \approx 32$, the value selected for the implementation described here. The search can be further hastened by declaring a mismatch immediately following the detection of the required number of discrepancies between search and data, instead of waiting for all n_{base} transitions.

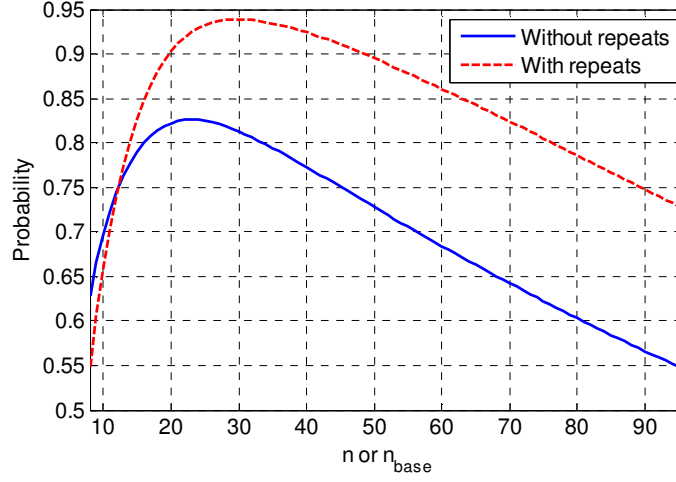


Figure 3.9: Peak probability in distribution of match/mismatch transition detection, with and without repeated averaging.

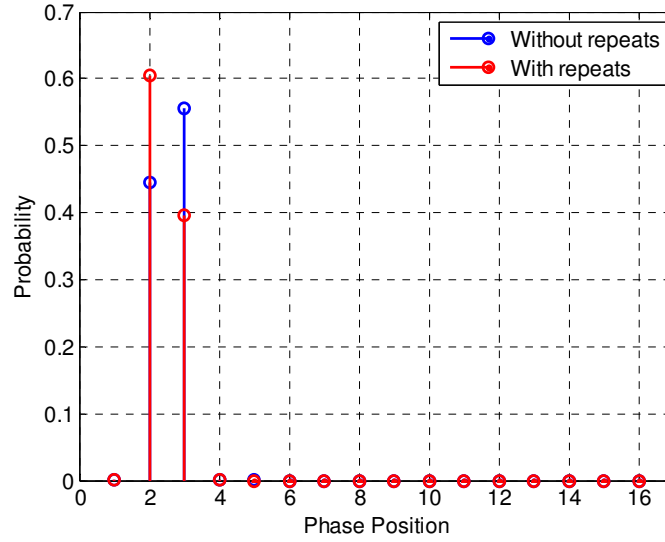


Figure 3.10: Probability of match-to-mismatch transition detection, $n = n_{base} = 128$.

Subsequent to the mismatch counter, an AND/OR filter is used to suppress the presence of subsidiary ‘false eyes’ that might result from reflections in the channel, crosstalk or large transient noise events. The previous k match/mismatch declarations are ANDed to eliminate spurious matches, and the output of the AND is then ORed to restore the original eye opening size (Figure 3.11). k defines the minimum eye-opening size the CDR is expected to track, and

should be set high enough to reject false eyes, but small enough to maintain sensitivity. Exact selection of k is not performance-critical, and a value of 4 is chosen for this implementation. The k -decision latency introduced by the AND/OR filter can be accounted for in the control logic.

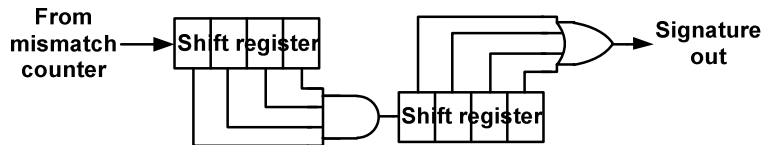


Figure 3.11: AND/OR filter with $k = 4$.

3.1.3 System startup and corner cases

The foregoing discussion of the ping-pong CDR algorithm assumes that the data clock is placed ‘correctly’, i.e. at a phase position such that the samples it produces are error-free estimates of the data that was transmitted. Obviously there is no guarantee that this is the case upon initial system startup; additionally, the CDR should be able to handle corner cases where the signature that has been collected does not provide a clean estimate of the eye opening (for example, it may only contain one match-to-mismatch transition, and therefore be missing one edge of the eye).

In order to handle these situations, the CDR control logic detects and handles two special-case signatures:

1. Only one match-to-mismatch transition detected (only one side of the eye is detected).
2. No match-to-mismatch transitions detected (eye is completely missing, or has size less than k and therefore been rejected by the AND/OR filter).

Case 1 occurs when the delay line is improperly calibrated, and is much shorter than 1 UI. In this case, one side of the match-to-mismatch transition will correspond to an open eye, while the other side will be where the eye is closed. The control logic will increase the delay line length and update the data clock so that it is at the mean of the detected match-to-mismatch transition and the end of the delay line corresponding to the open eye.

Case 2 occurs when the data clock is placed incorrectly, so comparisons with the search clock produce essentially random results with no clear eye opening. Lacking any information about the location of the eye opening, the control logic will simply increment the data phase by m phase

positions, where m is the largest integer that is smaller than k (the smallest eye opening expected) and not a factor of the overall number of phase positions available. This ensures that the control logic will sweep through the delay line in the smallest time possible while not skipping past the eye opening, and will (upon cycling through the delay line) eventually check all available phase positions. For this implementation, $k = 4$ and the delay line has 64 phase positions, so a value of 3 was chosen for m .

3.2 Shared CDR

The independent adjustability of each clock and the generalized sampling and re-timing paths of the ping-pong CDR allow it to be easily adapted to a shared multi-pin system. Instead of trading search/data clock functionality on a single pin, the search clock can ‘bubble’ through multiple pins (Figure 3.12). Only one pin is calibrated at a time, so only one extra clock is necessary. Thus, for N pins only $N + 1$ clocks are required, instead of the $2N$ required without sharing.

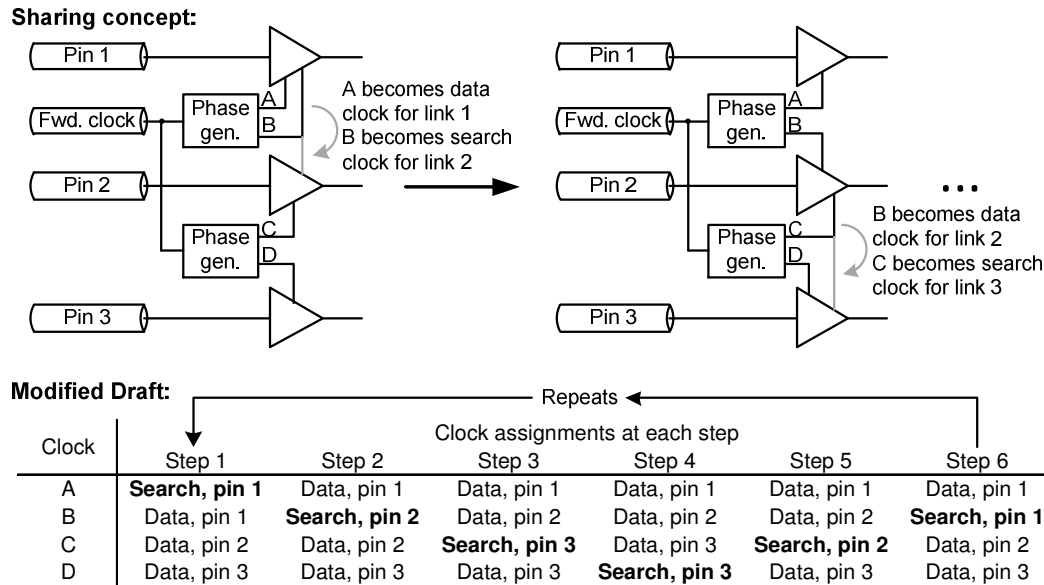


Figure 3.12: Sharing concept and modified draft algorithm, for 3 data pins and 4 clocks

The pins are calibrated in sequence, with the calibrated pin’s data clock swapping with the search clock at each step. The most equitable algorithm would be analogous to a standard draft;

each pin would be calibrated in sequence, from first to last, and the calibration would then return to the first pin. The time between calibrations of a particular pin would thus be N complete search and data phase update periods. However, the standard draft requires each clock to successively sample all of the pins, making its hardware cost prohibitive – it multiplies the number of input slicers required, complicates the input routing and calls for larger high-speed multiplexers to route the search and data samples. While this hardware overhead may be reasonable for small N , it does not scale well to large numbers of pins.

To avoid this constraint, a modified draft algorithm is used. Instead of returning to the first pin after the last pin has been calibrated, the modified calibration sweeps back-and-forth through the pins (Figure 3.12). It is less equitable and results in a $2N - 1$ period gap (in the worst case) between calibrations of any one pin, but requires each clock to sample no more than two adjacent pins and is therefore more hardware efficient, scaling well to large N . This draft scheme is well-suited for dense source-synchronous environments, where CDR bandwidth requirements are low and reducing hardware overhead is paramount.

3.3 Implementation

A block-diagram overview of the implemented system is presented in Figure 3.13. To save power and ease design constraints, the mismatch counter operates on quarter- and eighth-rate clocks, while the AND/OR filter, eye detection, clock phase placement and multiplexer control logic operates on a distinct low-frequency clock. All are synthesized from standard cells. The high-speed phase generation, slicing and multiplexing circuitry operates on a half-rate clock and is custom digital. As much as possible was implemented using static CMOS logic.

As the phase generator architecture used (described in the following sub-section) naturally generates clocks in pairs, and $N + 1$ clock phases are needed (one for each link's data clock, plus a bubbling search clock), an odd number of pins is called for. In this case, a three-link system is implemented to allow the performance of the shared CDR to be fully evaluated and extrapolated to wider links.

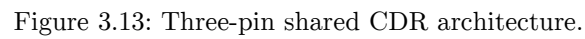


Figure 3.13: Three-pin shared CDR architecture.

3.3.1 Phase generator

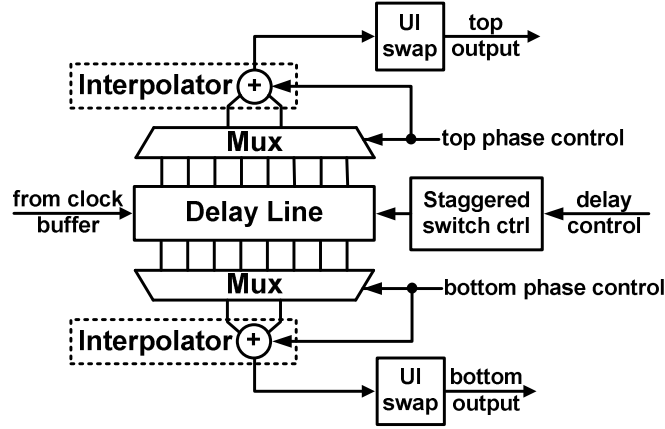


Figure 3.14: Phase generator architecture.

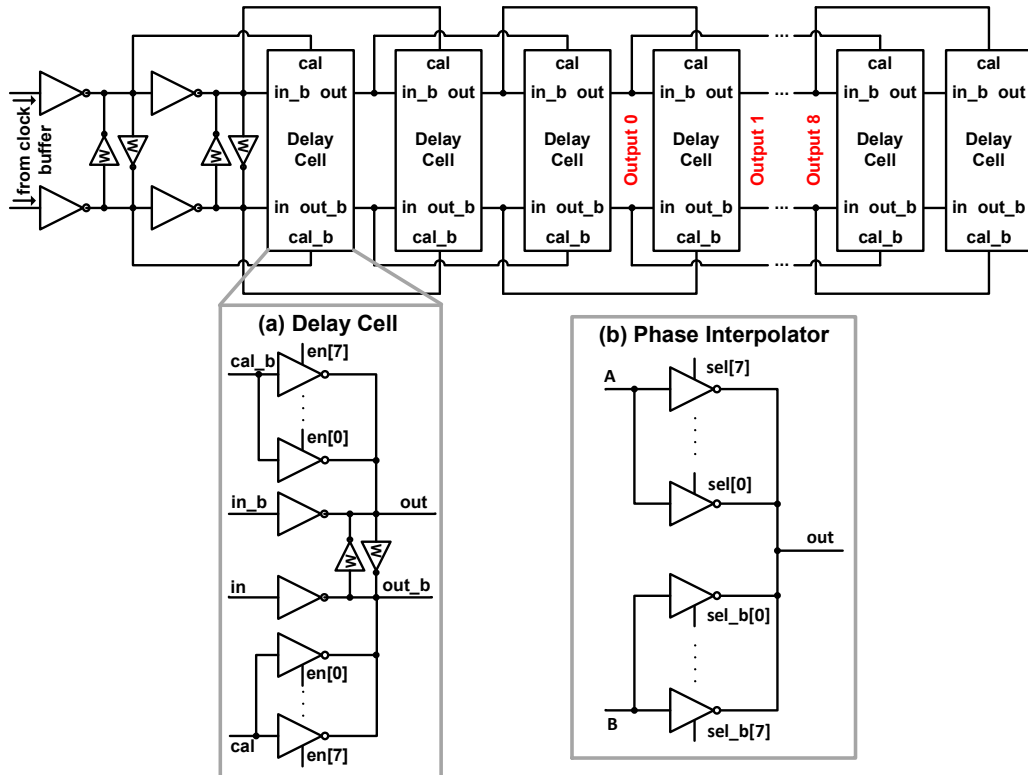


Figure 3.15: Delay line with (a) delay cell and (b) phase interpolator. Weak cross-coupled inverters are marked with a 'W'.

The core of the phase generator (Figure 3.14) is a direct digitally-modulated, differential delay line (Figure 3.15), with nine evenly-spaced output phases. Each cell of the delay line (Figure 3.15 (a)) is composed of tri-state buffers, which can be turned on or off to adjust the drive strength of each stage, thus the overall delay of the line [56]. Weak cross-coupled inverters are placed at the output of each cell to maintain phase alignment between the different differential paths and duty cycle. This scheme has the advantage of allowing an adjustable delay line implementation in pure static CMOS. However, the array of tri-state buffers and the wiring necessary to connect them imposes significant extra loading on the output of each delay cell, thus limiting the practical resolution of the delay adjustment. To overcome this drawback, the output of each delay cell is fed-forward to the calibration input (cal/cal_b in Figure 3.15) two cells away [57], thus reducing the size of the tri-state buffers necessary to achieve a large delay range.

An important consideration is the consistency of the phases of the output clocks when the delay line length is changed. These clocks are generated by interpolation of the outputs of a delay line; if the delay line length is changed abruptly, the phase of its outputs will likewise jump, thus causing a deviation in the phase of the generated clocks that increases jitter and could result in errors in CDR tracking. This effect is particularly severe when the outputs near the end of the delay line are being used to generate the clock, since the accumulated change in delay is largest at this point. To minimize this effect, the delay control code is updated in a stepwise manner, with hysteresis added to ensure that small changes in the number of phase positions per UI do not result in control code dithering. Additionally, the delay line is split into four sections of two delay cells each, with the delay updated section by section. Since the delay line can be updated slowly, updates to each section are staggered across several data phase updates; the phase effect of the delay update is therefore spread out, and the CDR only has to deal with it incrementally. In simulation, this staggering reduces the phase discontinuity per data phase update from 6 ps to 2.6 ps, or 0.7 phase positions at 9 Gb/s.

Two adjacent output phases of the delay line are selected via a multiplexer and interpolated to generate finer granularity. The phase interpolator itself (Figure 3.15 (b)) is composed of a pair of tri-state buffer arrays with shorted outputs; the interpolation ratio (α in Figure 2.13) is controlled by turning portions of this array on or off, while maintaining a constant total number

of active tri-states, thus ensuring a consistent output drive. Since there are 9 possible pairs of adjacent output phases from the delay line, and 9 settings of the phase interpolator, the complete phase generator has 64 total output phases, for an overall phase adjustment resolution of 6 bits.

3.3.2 Phase generator linearity

The linearity and resolution of the phase generator affects the final accuracy of the data clock placement by the CDR algorithm. Indeed, the phase generator is the only component of the system for which matching is performance-critical, so a detailed analysis of its linearity is worthwhile. Let Φ_1 and Φ_2 be the detected locations of eye edges 1 and 2 (as defined in Figure 3.4), respectively. Φ_1 and Φ_2 are the phase positions bounding the detected eye opening, and are used to determine the next data phase. The phase generator has limited resolution and could introduce nonlinearity, so there is some error in Φ_1 and Φ_2 relative the actual positions of eye edges 1 and 2 (Φ'_1 and Φ'_2):

$$\Phi_1 = \Phi'_1 + \epsilon_1, \quad \Phi_2 = \Phi'_2 + \epsilon_2 \quad (3.6)$$

where the worst-case error in terms of phase positions can be written by observing that it is affected by the resolution of the phase generator and its worst-case differential non-linearity (DNL):

$$\epsilon_1 = \epsilon_2 = \frac{1 + DNL_{\text{worst}}}{2} \quad (3.7)$$

The algorithm will place the next data phase at the average of the two phase positions:

$$\Phi_{\text{data}} = \frac{\Phi_1 + \Phi_2}{2} = \frac{\Phi'_1 + \Phi'_2 + \epsilon_1 + \epsilon_2}{2} \quad (3.8)$$

Finally, the placement of the data phase itself will be affected by the integral non-linearity (INL) between Φ_1 and Φ_2 , thus yielding an overall worst-case error, in terms of phase positions, of

$$\begin{aligned} \epsilon_{\text{data}} &= \frac{\epsilon_1 + \epsilon_2}{2} + INL_{\text{worst}} \\ &= \frac{1 + DNL_{\text{worst}}}{2} + INL_{\text{worst}} \end{aligned} \quad (3.9)$$

In order to minimize the data phase placement error, therefore, it is important to reduce the non-linearity of the phase generator. This can be broken down into the non-linearity of the phase generator's constituent components, the delay line and phase interpolator. First, consider the

DNL. Let L_x be the actual length of a LSB, L'_x be the ideal length of a LSB, Δ_x be the absolute worst-case DNL and δ_x be the worst-case DNL in LSBs, where $x = \{D, I, net\}$ for the delay line, the interpolator and the overall phase generator, respectively. The actual length of a delay line LSB can be written:

$$L_D = L'_D + \Delta_D \quad (3.10)$$

where

$$\delta_D = \frac{\Delta_D}{L'_D} \quad (3.11)$$

$$L_D = L'_D(1 + \delta_D) \quad (3.12)$$

Since there are 8 interpolator intervals per pair of delay line outputs, an LSB of the phase generator is:

$$\begin{aligned} L_{net} &= \frac{L_D}{8}(1 + \delta_I) \\ &= \frac{L'_D(1 + \delta_D)}{8}(1 + \delta_I) \end{aligned} \quad (3.13)$$

Now, the ideal phase generator LSB is simply $1/8^{\text{th}}$ the ideal delay line LSB:

$$L'_{net} = \frac{L'_D}{8} \quad (3.14)$$

$$\therefore L_{net} = L'_{net}(1 + \delta_{net}) = \frac{L'_D}{8}(1 + \delta_{net}) \quad (3.15)$$

Thus:

$$\begin{aligned} \frac{L'_D}{8}(1 + \delta_{net}) &= \frac{L'_D(1 + \delta_D)}{8}(1 + \delta_I) \\ 1 + \delta_{net} &= (1 + \delta_D)(1 + \delta_I) \\ \delta_{net} &= \delta_D\delta_I + \delta_D + \delta_I \end{aligned} \quad (3.16)$$

which suggests that both the delay line and interpolator contribute equally to the overall DNL of the phase generator. This formulation of the phase generator linearity assumes that the worst-case DNL of both the phase interpolator and the delay line coincide, which may not actually be the case; the linearity of the phase interpolator depends on the phase spacing of its inputs, so it is not constant across the delay line [34]. As a result, (3.16) should be viewed as a pessimistic estimate of the worst-case phase generator DNL.

A similar analysis can be conducted on the INL. Let Γ_x be the absolute worst-case INL and γ_x be the worst-case INL LSB's, where $x = \{D, I, net\}$ as before. The analysis of aggregate phase generator INL is not as straightforward as that for the DNL; consider the following set of cases where the delay line INL is always negative:

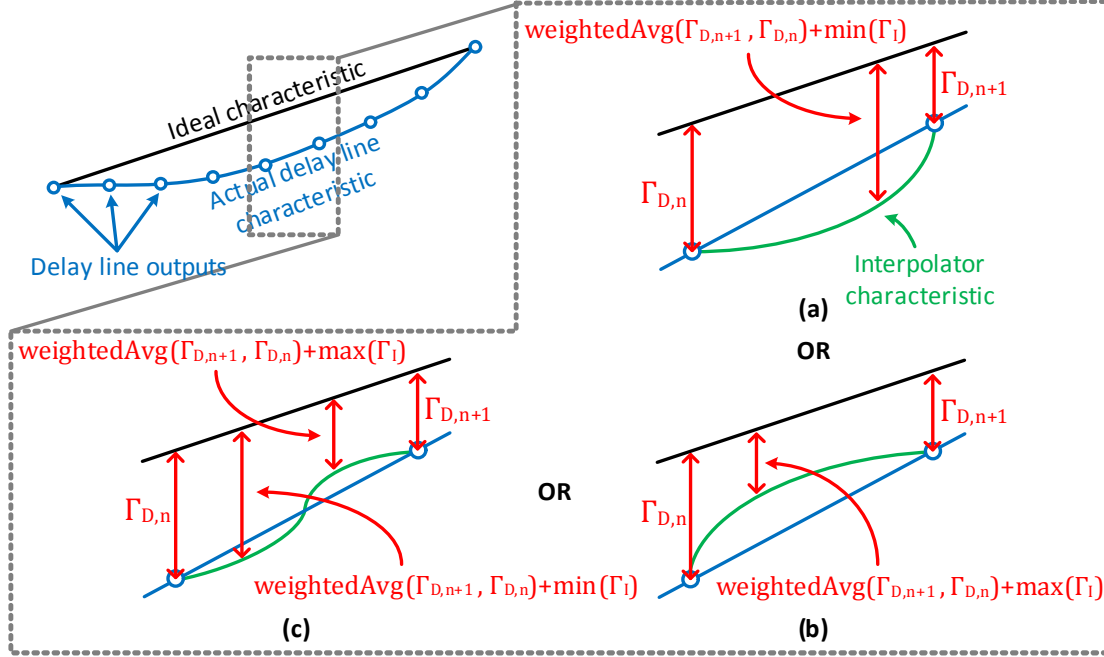


Figure 3.16: Delay line with negative INL and phase interpolator with (a) negative INL, (b) positive INL and (c) 'crossing' INL.

Notice that the worst-case net phase generator INL depends very strongly on the shape and severity (slope) of the phase interpolator and delay line characteristics. It is, however, possible to place bounds on the phase generator INL, since it is at least as bad as the delay line INL, and possibly as bad as the worst-case phase interpolator INL on top of that:

$$\Gamma_D < \Gamma_{net} < \Gamma_D + \Gamma_I \quad (3.17)$$

Since the delay line LSB is, ideally, 8 times the size of the phase generator LSB and the phase interpolator LSB is the same size as the phase generator LSB:

$$\begin{aligned} \frac{\Gamma_D}{L'_{net}} &< \frac{\Gamma_{net}}{L'_{net}} < \frac{\Gamma_D + \Gamma_I}{L'_{net}} \\ 8\gamma_D &< \gamma_{net} < 8\gamma_D + \gamma_I \end{aligned} \quad (3.18)$$

suggesting that the delay line INL is the dominant contributor to net phase generator INL.

Per (3.9), INL is a more significant contributor to data phase placement error than DNL, so the INL of the delay line is the most critical linearity. Several steps were taken to ensure good delay line matching. Although the linear nature of the delay line structure does not lend itself to common-centroid layout, extensive use was made of dummy devices. The wire parasitics presented to each delay stage were made as similar as possible (by using the shortest possible wire runs and copying them from stage to stage). Finally, pre-buffer stages were added to stabilize clock rise and fall times prior to the delay line proper.

Interpolator linearity requires that the phase difference between the input edges be less than half the edge rate (rise/fall time) of these inputs [34]. If the delay line is properly calibrated to 2 UI at the maximum target data rate of 9 Gb/s, each of the eight possible pairs of adjacent outputs has a phase spacing of about 28 ps. Due to the high fan-out of the buffers driving the interpolator (as shown in Figure 3.15 (b), each input buffer has to drive 8 tri-state buffers), it is relatively straightforward to ensure that input edge rates are faster than the 56 ps necessary to fulfil this linearity requirement.

3.3.3 Multiplexers

As the control logic is shared between multiple data pins, a key implementation challenge is the design of a high-speed multiplexer tree to route the search, data and clock signals of the pin-under-calibration to the control logic.

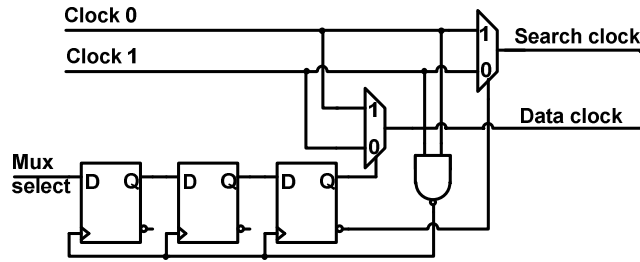


Figure 3.17: Search/data multiplexer (only clock routing shown; sample routing is similar).

The search/data multiplexers must be able to change the data clock of each pin without introducing errors. This is accomplished by delaying changes in the select signals until the input clocks to each search/data multiplexer are both high (Figure 3.17). This ensures that the swap is

made when no transition is occurring in either the clock or data inputs. However, the flip-flop storing the multiplexer state is clocked asynchronously with its input and may enter a metastable state. Since the flip-flop clock is generated from the overlap of two half-rate clocks, there is only a small window of time for metastability recovery. Therefore, a cascade of two synchronizers is used to give adequate metastability protection and ensure correct operation. The synchronizers introduce latency in the switch between clocks, but the control logic for the clock update runs at a much lower frequency, so this latency is inconsequential.

A further challenge is the long and asymmetric wiring run necessary to connect the search/data multiplexer of each pin and the pin-select multiplexer, which routes the search and data samples of the pin-under-calibration to the control logic. To ensure that proper timing is maintained between the recovered clock and data signals from each channel, they are co-routed and pipeline registers are inserted at the output of the search/data multiplexers.

3.3.4 Retiming logic

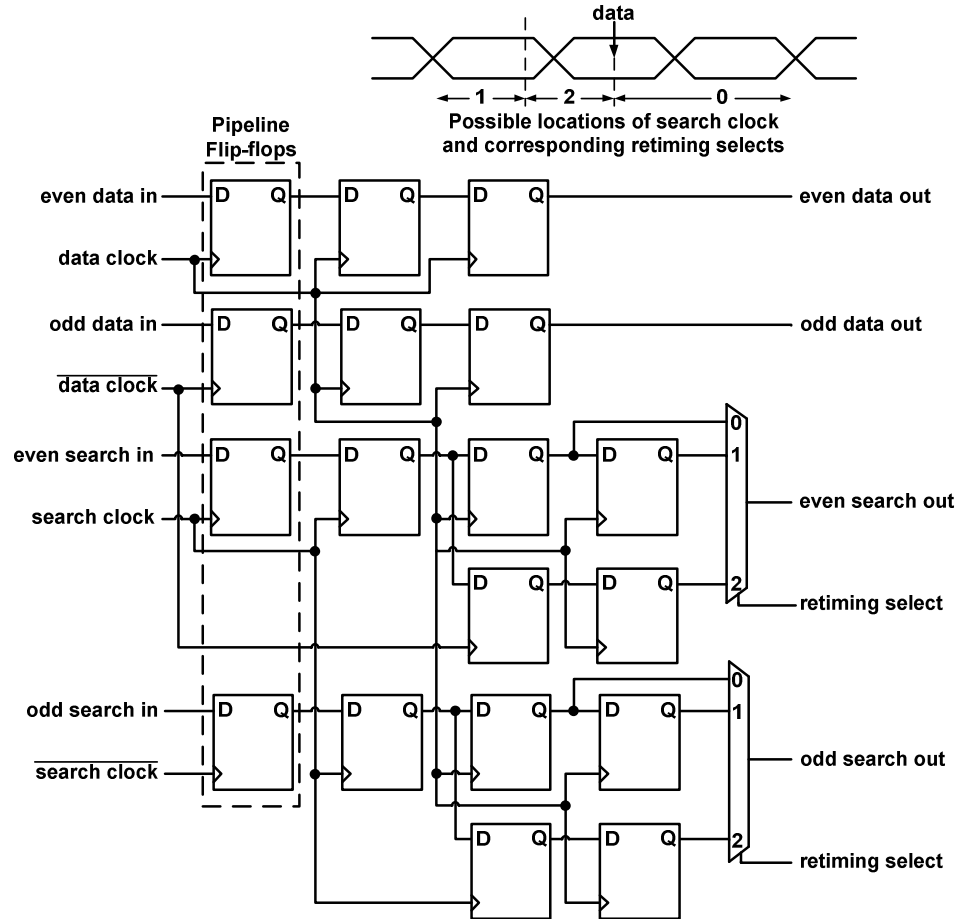


Figure 3.18: Retiming logic.

The mismatch counter needs to compare samples arriving from both the data and search clock. Since the search clock is at a varying (but known) phase offset from the data clock, the incoming samples need to be retimed before this comparison is made. This is accomplished through chains of flip-flops (Figure 3.18); as the phase offset varies from small ($1/32$ UI, a single phase step) to large (as much as 1.5 UI, depending on the search type and location of the data phase), each path in the retiming block is dedicated to a range of phase offsets. The appropriate path is selected based on the known location of the clocks.

The retiming logic takes samples from the odd phase of the clock to the even phase, and the timing for this transition is tight – it needs to complete in a full-rate instead of half-rate period.

To maximize the timing margin available, pipeline flip-flops (outlined in Figure 3.18) are added to the odd inputs, with the equivalent added to the even inputs for delay-matching purposes.

3.4 Hardware measurements

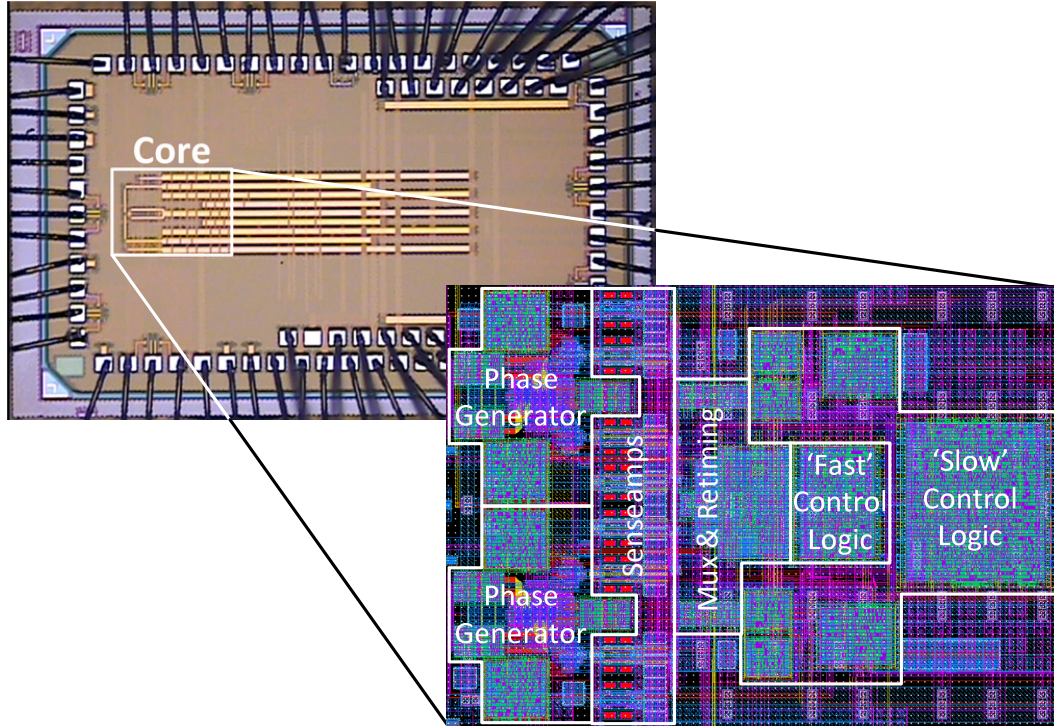


Figure 3.19: Die micrograph and core detail.

The CDR was fabricated in a 90 nm bulk CMOS process. The die micrograph and core detail are presented in Figure 3.19. Core area is $460\text{ }\mu\text{m} \times 330\text{ }\mu\text{m}$, in a $2.35\text{ mm} \times 1.45\text{ mm}$ die. Correct operation over an infinite delay range was verified by sweeping the input phase of each channel independently at data rates from 6 to 9 Gb/s. A 31-bit pseudo-random bit sequence (PRBS-31) input achieved $\text{BER} < 10^{-13}$.

Delay line response to calibration code was measured with a 4.5 GHz clock (i.e. data rate of 9 Gb/s), yielding a delay range of 183 to 278 ps. This corresponds to data rates between 7.2 Gb/s to 10.9 Gb/s (Figure 3.20), if the delay line is required to match 2 UI exactly. The CDR operated

correctly (with > 1 UI of jitter) as low as 6 Gb/s, well below this range. This confirms that exact delay line calibration is unnecessary for the eye-monitoring algorithm to function.

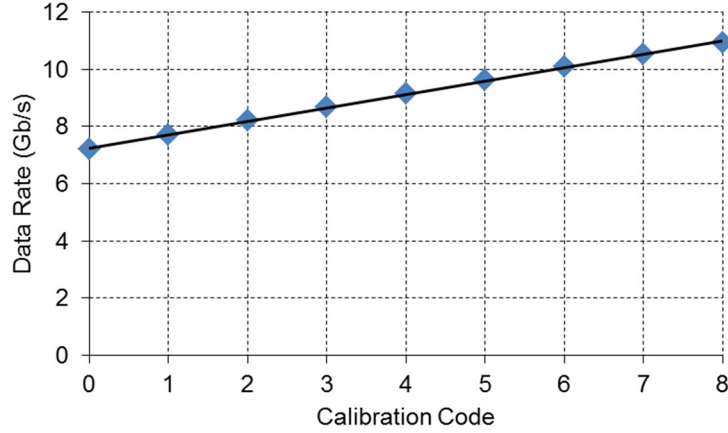


Figure 3.20: Data for 2 UI of delay, over the delay line calibration code range.

Phase generator linearity was measured, with a worst-case DNL of 0.44 LSB (where 1 LSB = 1 phase position), and a worst-case INL of 1.59 LSB (Figure 3.21). The delay line linearity can be isolated by selecting every 8th output of the phase generator, for which the phase interpolator has $\alpha = 0$. When this is done, the delay line shows a worst-case DNL of 0.22 LSB and worst-case INL of 0.18 LSB (Figure 3.22). Similarly, phase interpolator linearity can be isolated by looking at groups of 8 output phases; because the phase interpolator linearity varies depending on the exact phase spacing between its inputs, it depends on the linearity of the delay line. Figure 3.23 and Figure 3.24 plot the phase interpolator linearity across all 8 pairs of delay line outputs, showing a worst-case INL of 0.52 LSB and a worst-case DNL of 0.32 LSB.

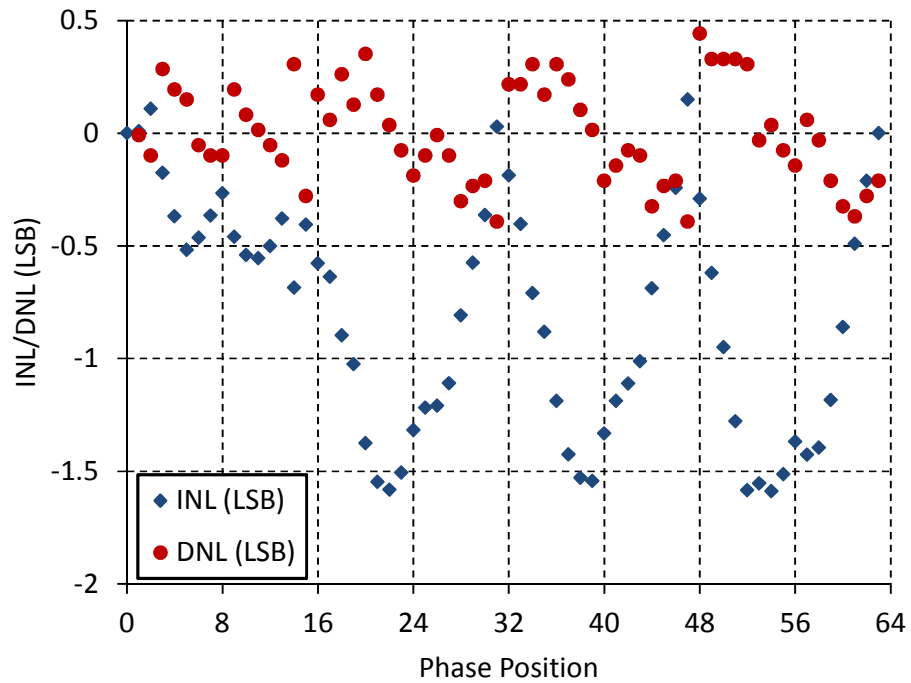
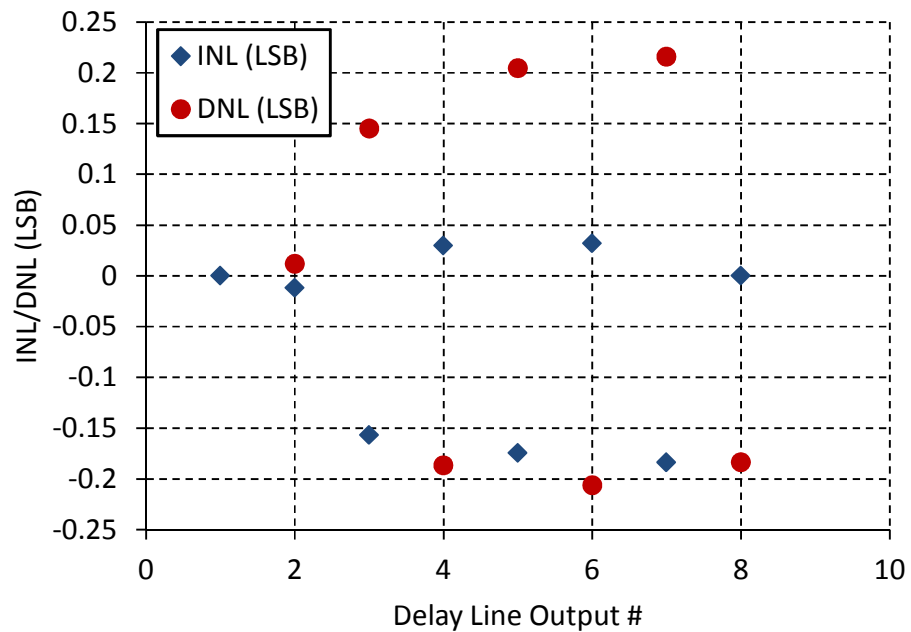


Figure 3.21: Phase generator non-linearity.

Figure 3.22: Delay line output non-linearity (note that the 0th output is missing, since this is inaccessible due to the configuration of the phase interpolator).

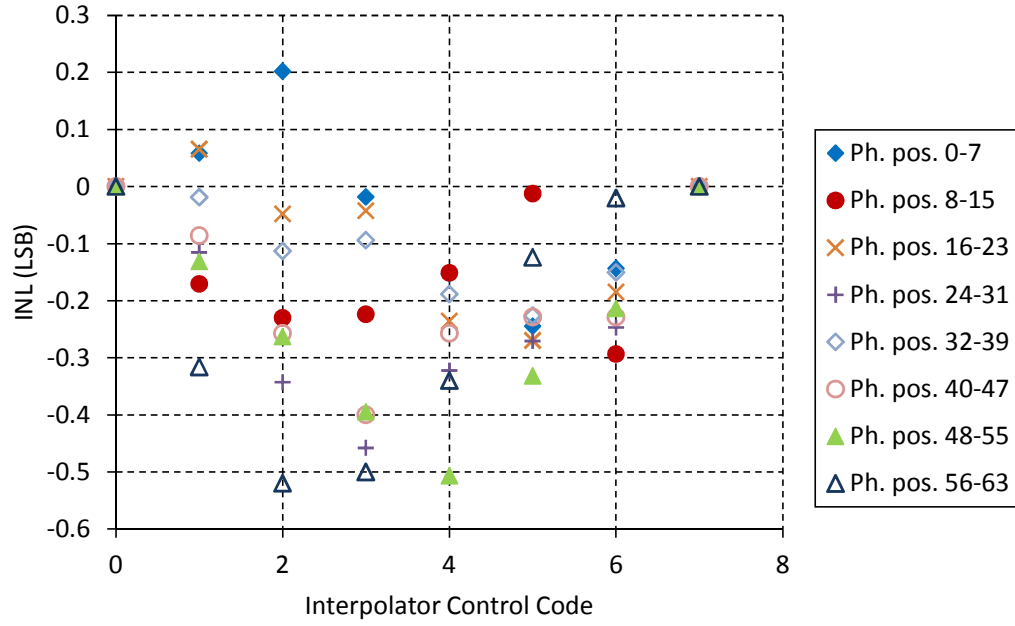


Figure 3.23: Interpolator INL, across different groups of 8 phase positions (each corresponding to one sweep through the interpolator).

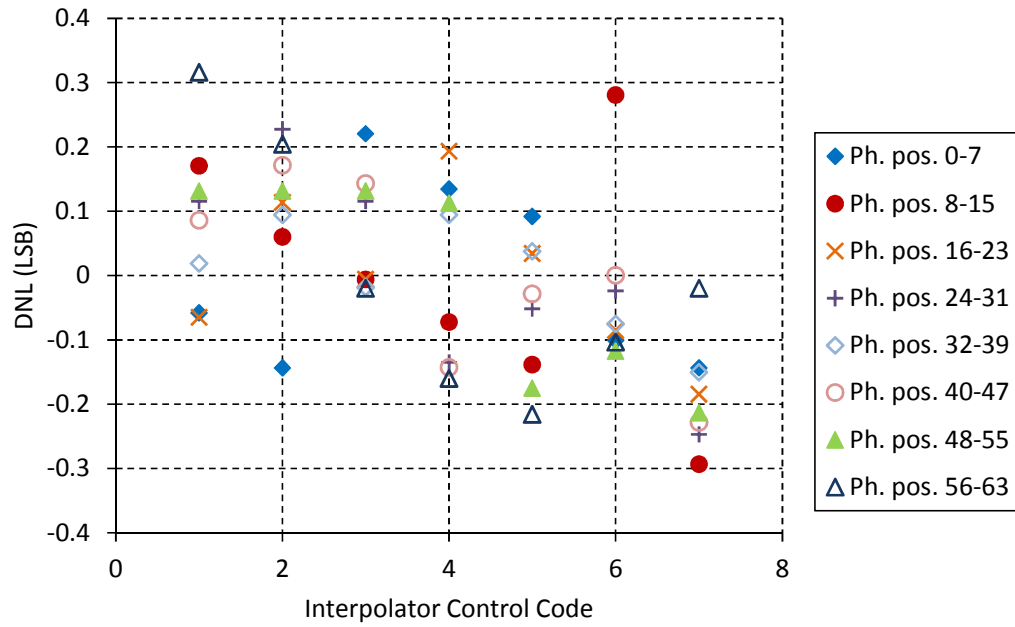


Figure 3.24: Interpolator DNL, across different groups of 8 phase positions (each corresponding to one sweep through the interpolator).

Using these results, it is possible to validate (3.16) and (3.18). (3.16) predicts a worst-case DNL (δ_{net}) of 0.61 LSB, which is somewhat greater than the measured DNL of 0.44 LSB – understandable, since, as described in the derivation itself, (3.16) is a pessimistic estimate. (3.18) predicts that worst-case phase generator INL is bound by $1.44 < \gamma_{net} < 1.96$ LSB, and the actual measured INL of 1.59 LSB is in this range. As expected, overall phase placement error is limited by the INL of the delay line.

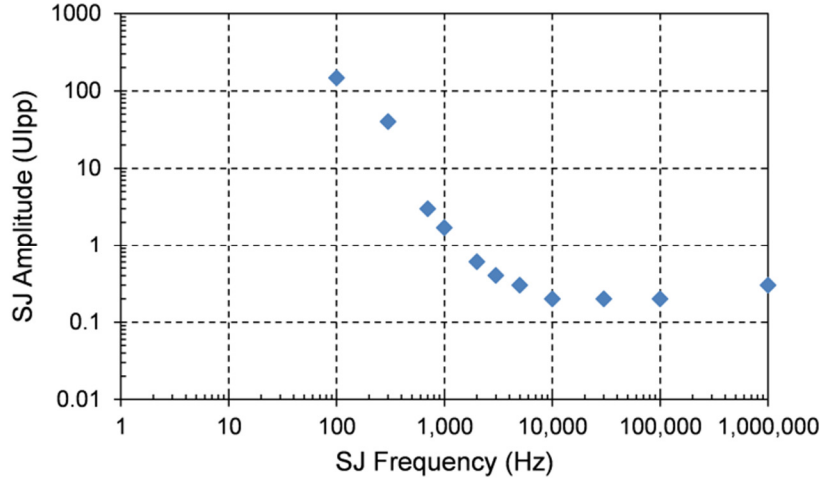


Figure 3.25: SJ tolerance with control logic clock at 40 MHz, for 3 x 9 Gb/s PRBS-7 input and $\text{BER} < 10^{-12}$.

Sinusoidal jitter (SJ) tolerance was measured with a control logic clock of 40 MHz (Figure 3.25). The period between data clock phase updates is limited primarily by the speed of the control logic, so an almost directly proportional relationship exists between the frequency offset tolerance (equivalently, the SJ tolerance bandwidth) and the control logic clock frequency. This is confirmed by measured results up to 50 MHz (limited by the design of the control logic that emphasized low-power operation at the expense of speed), which match simulated results closely (Figure 3.26). Cycle-accurate Verilog-AMS simulations at faster clocks show that a linear relationship is maintained up to at least 625 MHz. This suggests a direct trade-off between system performance and control logic power consumption; since the presented implementation is source-synchronous, low CDR bandwidths are tolerable and control logic power consumption is prioritized by targeting a lower clock frequency. Higher performance can be achieved by targeting

a faster control logic clock, allowing the CDR to calibrate plesiochronous links with small frequency offsets.

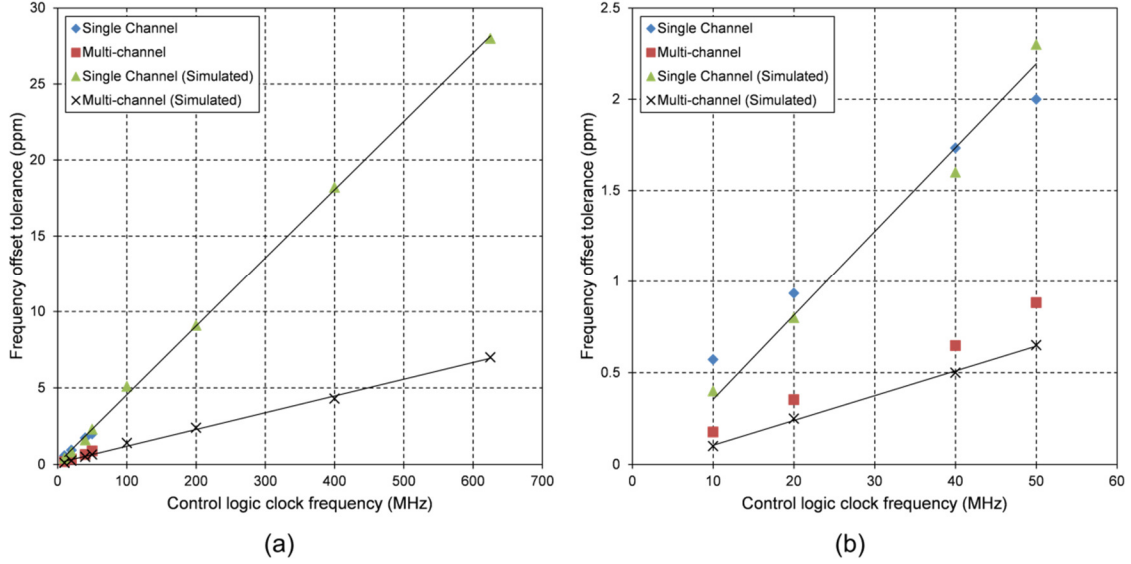


Figure 3.26: (a) Frequency offset tolerance scaling for 3 x 9 Gb/s PRBS-7 input (measured BER $< 10^{-12}$ and simulated BER $< 10^{-6}$), with (b) low frequency detail.

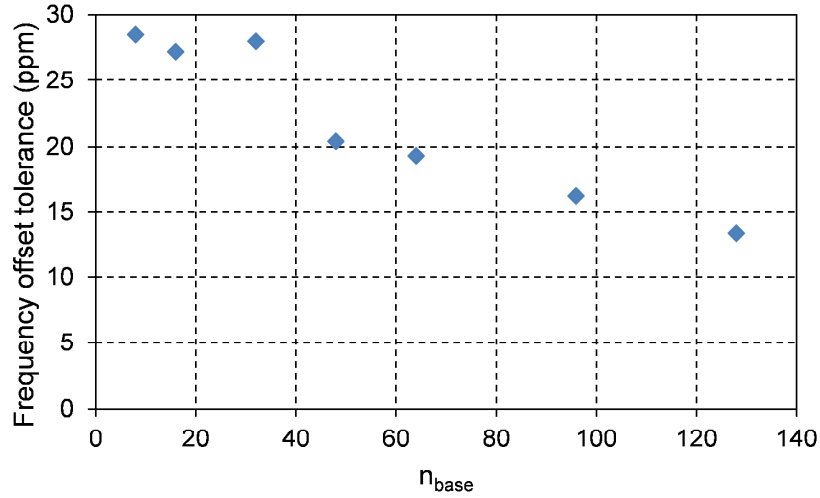


Figure 3.27: Effect of n_{base} on frequency offset tolerance, simulated on a single channel at 625 MHz with PRBS-7 input and BER $< 10^{-6}$.

The filtering parameter n_{base} also has a significant effect on CDR bandwidth, as described in Section 3.1.2. Frequency offset tolerance was simulated at different values of n_{base} , using the

highest logic frequency (625 MHz) to minimize the effect of logic delays (Figure 3.27). $n_{base} < 32$ results in faster searches and more frequent data phase updates, but CDR bandwidth is not improved since gains in speed are offset by a decrease in eye detection accuracy, as described in Section 3.1.2 (specifically, see Figure 3.9). $n_{base} > 32$ slows the search process in addition to degrading eye detection accuracy, so bandwidth decreases. These results validate the choice of $n_{base} = 32$ indicated by the theoretical analysis.

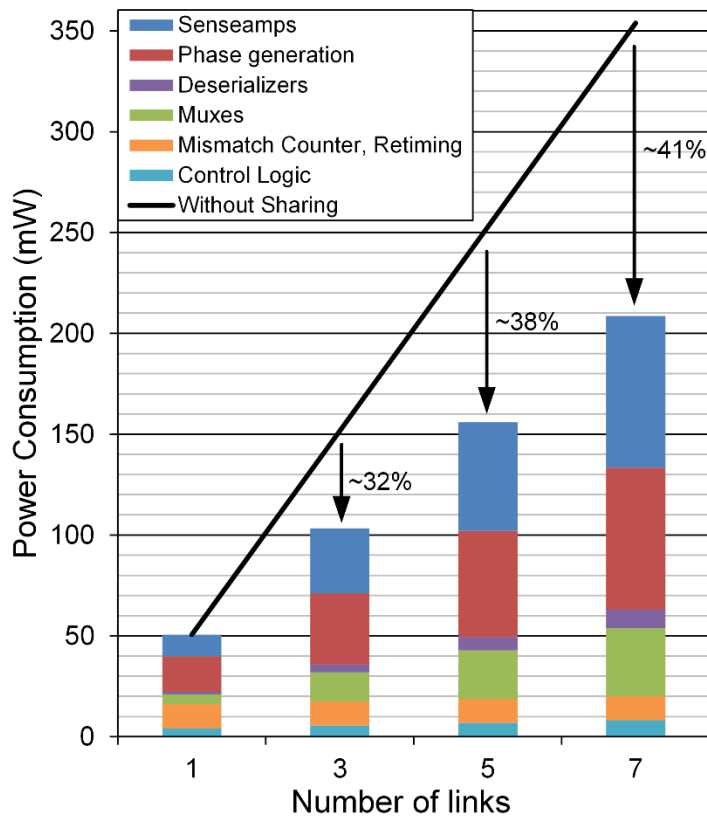


Figure 3.28: Power breakdown and scaling performance

Overall power consumption of the 3-pin system, operating at 9 Gb/s, is 103.3 mW, or 3.8 pJ/b. Operation at 6 Gb/s allows a slight reduction in the supply voltage and yields an overall power consumption of 45.6 mW, or 2.5 pJ/b. A module-by-module breakdown of power consumption was inferred by scaling measured results using simulation data (Figure 3.28). By reducing the number of clocks required, the shared CDR brings the phase generation power consumption in-line with that of the input slicers (senseamps), the next most significant

component. Sharing the high-speed re-timing and mismatch counter logic, as well as the low-speed control logic, helps minimize control logic overhead at the expense of expanding multiplexer complexity. Further optimization is possible. As a simple example, this implementation keeps all the input slicers running all of the time. Since the slicers exist in pairs of sets per pin (one set each for search and data), and only the data set is required unless the pin is being calibrated, it is possible to further reduce power consumption by disabling the clocks to the unused slicers. In the three-pin implementation, input slicers consume about 31% of the overall power. Adding clock-gating would allow 2 of the 6 sets of slicers to be disabled, potentially achieving an overall system power savings of about 10%.

Even without such optimizations, the three-pin implementation uses about 32% less power than a naïve tripling of the single-pin system. Further scaling benefits can be realized by extending the system to wider links. The amount of sharing would ultimately be limited by the width of the channel-select multiplexer, the routing to this multiplexer and/or the desired jitter and frequency offset tolerance. It is possible to control multiplexer complexity by repeating the re-timing and mismatch counter over several subsets of pins in the overall link, and to mitigate performance loss due to sharing by increasing the control logic clock, although both these approaches sacrifice some of the power efficiency of the shared system. Overall performance of the system is summarized in Table 3.1.

Process	90 nm bulk CMOS
Data Rate (Gb/s)	3 x 9
Supply Voltage (V)	0.9 / 1.2 (6 Gb/s) 1.0/1.25 (9 Gb/s)
Power (mW)	45.7 (6 Gb/s) 103.34 (9 Gb/s)
Area (mm ²)	0.15
Power FOM (pJ/b)	2.5 (6 Gb/s) 3.8 (9 Gb/s)
Area FOM (mm ² /Gb/s)	0.008 (6 Gb/s) 0.006 (9 Gb/s)

Table 3.1: Performance summary, 3 x 9 Gb/s ping-pong CDR

3.5 Equalization adaptation

Besides clock recovery and synchronization, the eye-monitoring nature of the ping-pong CDR lends itself to equalization adaptation, particularly with non-traditional DFE types for which the traditional sign-sign least-mean-squares (SS-LMS) adaptation is not applicable. The ping-pong CDR-based equalization adaptation is especially useful in on-chip links; since these tend to be highly-parallel and mesochronous, bandwidth is not a concern and the CDR can easily realize the power and area advantages of sharing. Additionally, the ping-pong CDR's all-digital nature makes it easy to embed in the midst of large digital systems, such as multi-core processors, where long (>1 mm) on-chip links are frequently encountered.

3.5.1 On-chip links

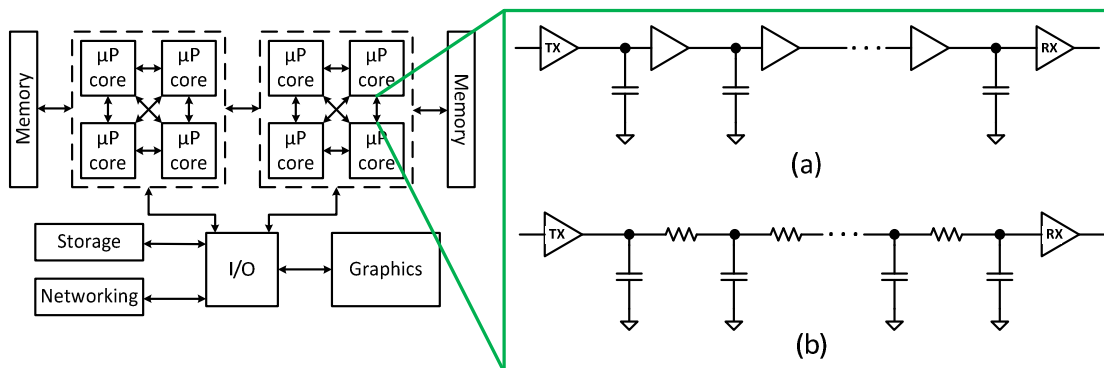


Figure 3.29: On-chip links, within the context of a two-socket server. Individual CPU chips indicated by dashed lines. (a) repeated, full-swing and (b) RC-limited, low-swing links shown.

A brief discussion of on-chip links is helpful to set the context. Long on-chip wires are very lossy, due to the large amount of resistance and capacitance that they experience. The traditional approach to maintaining signal integrity across such a channel is to split it up and insert repeaters every so often (Figure 3.29(a)), thus limiting the RC load imposed by any single segment of the wire and maintaining full signal swing across it [58]. This method is, however, power- and area-inefficient; each driver obviously requires a certain amount of power, and in modern digital systems that incorporate dynamic voltage and frequency scaling (DVFS) for power

efficiency, the repeaters frequently require their own supply rail to maintain correct operation even as the supply voltages of the source and destination (e.g. two microprocessor cores) change.

In order to work around this, several alternatives have been proposed that reduce the number of repeaters required [59], [60]. However, full-swing links are fundamentally limited by the relationship between power and voltage [58]:

$$P \propto CV^2f \quad (3.19)$$

where C is the capacitance being driven and f is the toggling rate. As voltage scaling has slowed, the power required to drive the wire alone will continue to increase untenably as the required signalling density (thus, wire capacitance) and data rate rise in future digital systems.

Reducing on-chip communication power, then, requires reduction of the signal swing so that it is less than the supply voltage. Several techniques for low-swing interconnect have been proposed, including those that utilize an RF-like transmission line approach [61], [62], or RC-limited channels [63], [64]. Implementation of on-chip transmission lines is challenging, since densely-packed metal wires face high loss and strong coupling to the surrounding metal. In order to reduce these parasitics and achieve a sufficient amount of inductance, transmission line implementations have to resort to wide-pitch channels, and the use of many layers of the metal stack to provide adequate separation between the ground plane and signal line. Furthermore, the RF-like approaches require the generation of high-frequency signals, either by upconversion [61] or the deliberate generation of high-frequency harmonics [62]. Both of these techniques are power-hungry and require use of relatively sophisticated analog processing, posing integration challenges in large digital systems.

Conversely, the RC-limited approach seeks to maximize the bandwidth density (measured in Hz/ μm of wire pitch) of the communications link, accepting that this may mean that individual wires have high loss. The bandwidth density (β) of a link with a structure similar to Figure 3.31 can be approximated in terms of the Elmore delay (τ) of its constituent wires and their pitch (P):

$$\beta \approx \frac{1}{2\pi \cdot \tau \cdot 2P} \quad (3.20)$$

In calculating the Elmore delay, it is traditional to split the wire up into N equal Π segments (as in Figure 3.29(b)), then calculate it as:

$$\tau = \left[\sum_{i=1}^{N-1} \frac{i \cdot R_{wire}}{N} \cdot \frac{C_{wire}}{N} \right] + R_{wire} \cdot \frac{C_{wire}}{2N} + R_{drive} \cdot C_{wire} + (R_{drive} + R_{wire})C_{load} \quad (3.21)$$

Where R_{drive} is the effective resistance of the transmitter driving the wire, C_{load} is the load capacitance presented by the receiver, and R_{wire} and C_{wire} are the overall resistance and capacitance of the wire, respectively. For long, thin on-chip wires $R_{wire} \gg R_{drive}$ and $C_{wire} \gg C_{load}$, so (3.21) reduces to:

$$\begin{aligned} \tau &\approx \left[\sum_{i=1}^{N-1} \frac{i \cdot R_{wire}}{N} \cdot \frac{C_{wire}}{N} \right] + R_{wire} \cdot \frac{C_{wire}}{2N} \\ &= \frac{R_{wire} \cdot C_{wire}}{N^2} \left[\sum_{i=1}^{N-1} i \right] + \frac{R_{wire} \cdot C_{wire}}{2N} \\ &= \frac{R_{wire} \cdot C_{wire}}{N^2} \cdot \frac{(N-1)(N)}{2} + \frac{R_{wire} \cdot C_{wire}}{2N} \\ &= \frac{R_{wire} \cdot C_{wire}}{2} \end{aligned} \quad (3.22)$$

Using this equation in (3.20) yields:

$$\beta \approx \frac{1}{\pi R_{wire} C_{wire} P} \quad (3.23)$$

The resistance of the wire can be straightforwardly estimated as:

$$R_{wire} \approx R_{\square} \cdot \frac{l}{w} \quad (3.24)$$

Where R_{\square} is the resistance-per-square of the wire, l is its length and w is its width. C_{wire} is composed of several different components:

$$C_{wire} \approx (C_{top} + C_{bottom} + 2 \cdot C_C) \cdot l \quad (3.25)$$

The complexity of the wire structure makes a purely analytical approach to modelling wire capacitance less than useful – attempts to do so produce equations that are too complex for useful computation and do little to enhance understanding of the underlying phenomena. Instead, the accepted approach is to separate the effects of the various contributing components (such as parallel-plate fields and fringing fields) as far as possible, and use curve-fitting to empirical data to derive the parameters for these contributions [65]. For example, for C_{top} (the capacitance-per-unit-length to the upper metal layer, which is modelled as a ground plane):

$$C_{top} \approx \epsilon_{top} \left(\frac{w}{s_{top}} + 2.04 \left(\frac{s_{side}}{s_{side} + 0.54 \cdot s_{top}} \right)^{1.77} \left(\frac{h}{h + 4.53 \cdot s_{top}} \right)^{0.07} \right) \quad (3.26)$$

where $\epsilon_{top} = \epsilon_0 K_{top}$ is the permittivity of the dielectric to the upper ground plane, s_{top} is the height of this dielectric, h is the height of the wire and $s_{side} = P - w$ is the spacing between wires. In a similar fashion,

$$C_{bottom} \approx \epsilon_{bottom} \left(\frac{w}{s_{bottom}} + 2.04 \left(\frac{s_{side}}{s_{side} + 0.54 \cdot s_{bottom}} \right)^{1.77} \cdot \left(\frac{h}{h + 4.53 \cdot s_{bottom}} \right)^{0.07} \right) \quad (3.27)$$

Finally, C_C is the coupling capacitance to the neighbouring (grounded) wires:

$$C_C \approx \epsilon_{side} \left(1.41 \frac{h}{s_{side}} \exp \left(-\frac{4s_{side}}{s + 8.01s_{avg}} \right) + 2.37 \left(\frac{w}{w + 0.31s_{side}} \right)^{0.28} \cdot \left(\frac{s_{avg}}{s_{avg} + 8.96s_{side}} \right)^{0.76} \exp \left(\frac{-2s_{side}}{s_{side} + 6s_{avg}} \right) \right) \quad (3.28)$$

where $s_{avg} = (s_{top} + s_{bottom})/2$. For metal 7 in a typical 9-metal 65 nm process, the foregoing analysis yields a bandwidth density optimum when $P \approx 0.7 \mu\text{m}$ and $w \approx 0.4 \mu\text{m}$ (Figure 3.30).

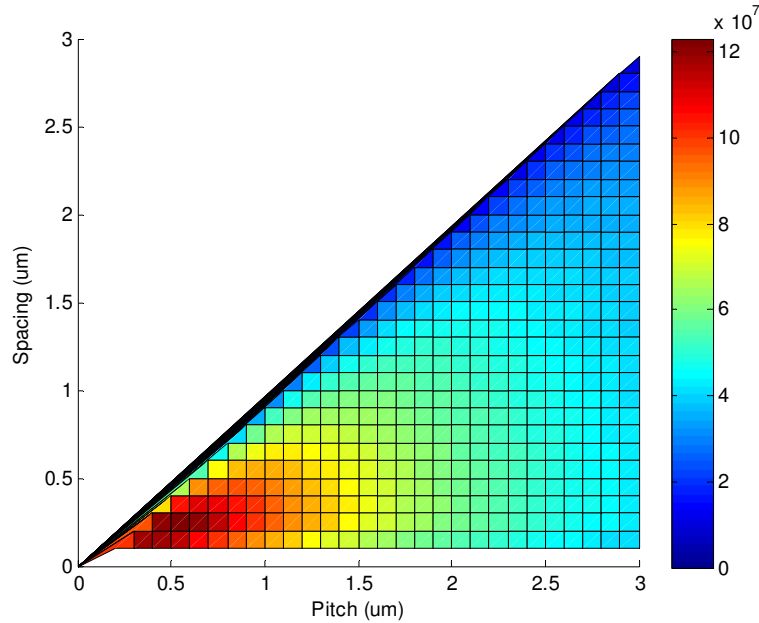


Figure 3.30: Bandwidth density (Hz/um) estimated over wire pitches up to 3 um in metal 7 of a typical 9-metal 65 nm process

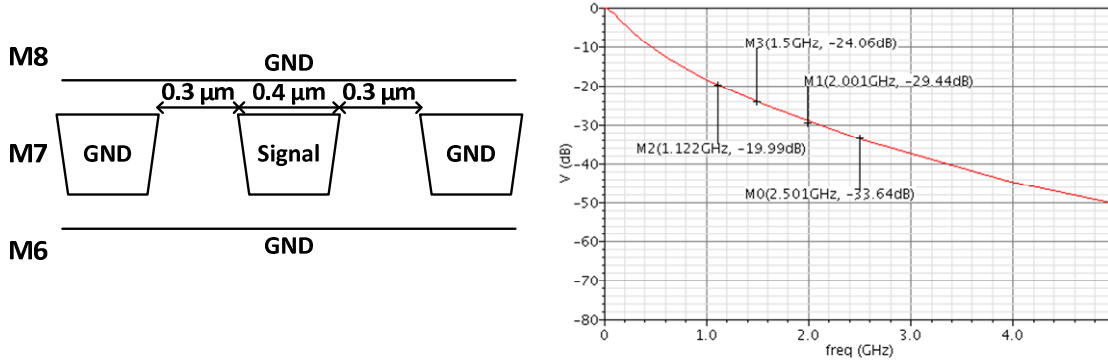


Figure 3.31: Bandwidth density-optimized on-chip channel in typical 9-metal 65 nm process, with frequency response for 10 mm length.

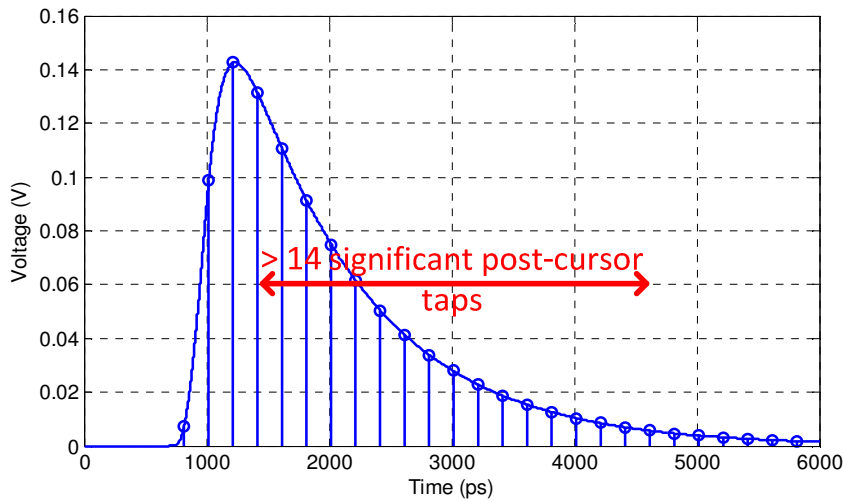


Figure 3.32: Pulse response of channel in Figure 3.31, at 5 Gb/s.

Although optimized for good bandwidth *density*, the bandwidth of individual wires is quite low, and they experience high channel loss (e.g. 33.6 dB @ 2.5 GHz in Figure 3.31). Equalization is necessary to communicate at high data rates over such channels; however, the large number of equalization taps required to overcome the long-tail ISI produced by an RC-limited channel (Figure 3.32) renders traditional transmitter pre-emphasis and DFE inefficient. Instead, a continuous time IIR analog filter can be used to generate the feedback signal for a DFE. In the case of an RC-limited on-chip channel, the channel response is simple and can be emulated with a simple first-order RC filter [66], [41] (Figure 3.33).

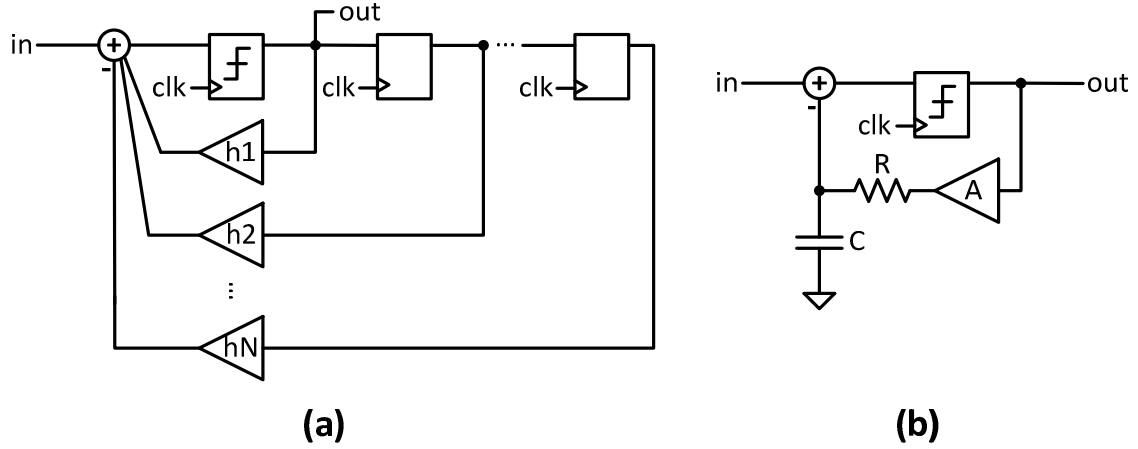


Figure 3.33: (a) Classic DFE with discrete-time FIR feedback, and (b) DFE with continuous-time RC feedback, replacing all N taps in the FIR feedback.

While providing compelling power benefits, a DFE using IIR feedback poses an adaptation challenge. The passive components of the feedback filter are subject to considerable process variation in deep sub-micron CMOS, as are the wire channels themselves. In many-core systems, with hundreds or thousands of such links, it becomes infeasible to include sufficient design margin to account for these variations, or to provide post-fabrication correction to tune this variability away. Instead, on-chip adaptation is required to meet this challenge. Conventional SS-LMS techniques used for FIR-feedback DFE adaptation rely on the fact that each tap weight independently controls the residual ISI at the corresponding post cursor (e.g. tap h_3 affects the residual ISI of the 3rd post-cursor) [33]. Since such a relationship does not exist when IIR-feedback is used (e.g. in Figure 3.33(b), R and C affect the time constant of the filter, and therefore the residual ISI over many post-cursors), alternative means of adaptive equalization need to be found. For instance, [67] proposes a pattern-guided method that attempts to minimize ISI by isolating the known contributions from pre-determined bit sequence. However, this approach does not directly cancel the first (most significant) post-cursor, nor does it account for ISI contributions beyond the 3rd post-cursor.

3.5.2 Eye-monitor-based adaptive equalization

The key limitation with pattern-guided approaches to adaptive IIR equalization such as [67] is that they need to store and detect long patterns of bits in order to sense long-tail ISI. As the pattern depth increases, adaptation decision logic also gets more complex. Using the eye-monitor information from the ping-pong CDR avoids this problem, since the eye information is collected without relying on specific patterns of bits in the incoming signal. In the course of operation, the ping-pong CDR detects the edges of the eye in order to place the data clock. It is a relatively simple matter to use this eye-edge information to determine the size of the eye opening, and this information can be filtered and used to adapt IIR feedback filter parameters for the optimum eye opening size. As a result, the overhead for adding equalization adaptation is minimal – a small amount of extra low-frequency logic that can be designed in HDL and synthesized into standard cells.

The ping-pong CDR can handle small frequency offsets; if a frequency offset is present, it will distort the eye size estimate by causing one edge of the eye to drift after the other has already been found. This distortion can be compensated for by estimating the amount and direction of the frequency offset. Let u be the time (in UIs) between UI swaps. Assuming the CDR is accurately tracking the center of the eye, the time between UI swaps (see Figure 3.5) is approximately the time taken for the frequency offset to cause a single UI worth of drift, so the actual clock period can be written:

$$T_{clock} = 1 \mp \frac{1}{u} = \frac{u \mp 1}{u} \text{ UI} \quad (3.29)$$

where the sign depends on the direction of the frequency offset (positive frequency offset implies shorter T_{clock} and vice versa). This can be re-arranged to yield the actual clock frequency:

$$f_{clock} = \frac{1}{T_{clock}} = \frac{u}{u \mp 1} \text{ UI}^{-1} \quad (3.30)$$

So the frequency offset is:

$$\Delta f = \frac{f_{clock} - f_{nom}}{f_{nom}} = \frac{u/u \mp 1 - 1}{1} = \frac{\pm 1}{u \mp 1} \text{ UI}^{-1} \quad (3.31)$$

where f_{nom} is the nominal clock frequency. Note that the foregoing derivation is for a full-rate clock. In a half-rate system, the clock is twice as long as a UI, so (3.29) becomes:

$$T_{clock} = 2 \mp \frac{1}{u} = \frac{2u \mp 1}{u} \text{ UI} \quad (3.32)$$

$$\Rightarrow \Delta f = \frac{f_{clock} - f_{nom}}{f_{nom}} = \frac{u/2u \mp 1 - 1/2}{1/2} = \frac{\pm 1}{2u \mp 1} \text{ UI}^{-1} \quad (3.33)$$

As an example, consider a half-rate system where there is a +50 ppm frequency offset. This implies that:

$$\begin{aligned} 50 \text{ ppm} &= \frac{1}{2u - 1} \\ \Rightarrow u &\approx 10,000 \text{ UI} \end{aligned} \quad (3.34)$$

When properly calibrated, the phase generator of the ping-pong CDR has 64 phase positions over 2 UI. Therefore, 1 UI \approx 32 phase positions. To determine the amount of frequency offset correction that needs to be applied, the control logic can count the number of UIs between UI swaps, and scale this value by the number of phase positions per UI (i.e. 32) and the time between detections of the eye edges (m , in UIs):

$$k_{corr} = \frac{d \cdot 32 \cdot m}{u} \text{ phase positions} \quad (3.35)$$

where d indicates the polarity of the correction to be applied, and is determined by:

$$d = \begin{cases} -1, & \Delta f < 0 \wedge (\text{no UI swap} \vee \text{UI swap to lower phase position}) \\ & \vee (\Delta f > 0 \wedge \text{UI swap to higher phase position}) \\ 1, & \Delta f > 0 \wedge (\text{no UI swap} \vee \text{UI swap to lower phase position}) \\ & \vee (\Delta f < 0 \wedge \text{UI swap to higher phase position}) \end{cases} \quad (3.36)$$

As an example, consider a half-rate system where the eye opening, ϵ , is 24 phase positions (i.e. 3/4 UI) and the frequency offset is 50 ppm. The time between detection of the two eye edges can be approximated as:

$$m = (\epsilon + 2k + 2) \cdot \rho \cdot n_{base} \text{ UI} \quad (3.37)$$

where k is the size of the AND/OR filter, n_{base} is the averaging period used by the mismatch counter, and ρ is the average transition density (see section 3.1.2 for a discussion on these parameters). This approximation assumes that the mismatch counter repeats itself once at each eye edge. Using $k = 4$, $\rho = 0.5$ and $n_{base} = 32$, this yields $m = 544$. From (3.34), $u \approx 10,000$. Therefore, the magnitude of the applied correction factor will be:

$$|k_{corr}| = \frac{32 \cdot 544}{10,000} \approx 1.75 \approx 2 \text{ phase positions} \quad (3.38)$$

Note that (3.37) ignores the effect of control logic overhead; if the logic is operating at a much lower frequency than the data rate in order to save power, m could well be considerably larger, increasing the amount of correction required.

The eye-monitor-based adaptive equalization procedure is split into three basic stages:

- **Initial search:** the eye is assumed to be closed, and the filter parameters are sequentially swept until the eye-monitor detects an open eye.
- **Guided adaptation:** when an open eye is found, the detected eye size is used to guide the stepping of filter parameters until an optimum has been found.
- **Converged monitoring:** once a presumed optimum has been reached, filter parameters are frozen. The detected eye size is monitored, and if it falls below a set threshold, the adaptation re-enters the guided adaptation stage.

The algorithm is run each time the ping-pong CDR generates a new eye opening estimate (i.e. when it updates the data clock), and may be described:

Algorithm 3.1: Eye-monitoring Adaptive Equalizer

Inputs:

x_0, x_1 // edges of the eye, found by ping-pong CDR
 k_{corr} // frequency offset correction (see above)

Outputs:

$H = (h_1, h_2 \dots h_n)$ // filter parameters

Persistent variables (and initial states):

$y_{accum} \leftarrow 0$ // accumulated eye sizes
 $z_{tgt} \leftarrow 0$ // target average eye size
 $i \leftarrow 1$ // current filter parameter ($\in \{1, 2 \dots n\}$)
 $j \leftarrow 0$ // averaging counter
 $H_{dir} = (h_{1,dir} \dots h_{n,dir}) \leftarrow (+1 \dots +1)$ // parameter adjustment direction ($\in \{-1, +1\}$)
 $allConverged \leftarrow false$ // flag; indicates filter parameters converged

Parameters:

$H_{max} = (h_{1,max} \dots h_{n,max})$ // maximum possible values of H
 $H_{min} = (h_{1,min} \dots h_{n,min})$ // minimum possible values of H
 $H_{step} = (h_{1,step} \dots h_{n,step})$ // step size for each filter parameter
 y_{min} // minimum valid eye size
 y_{thresh} // back off current filter parameter if current
// eye size falls below $z_{prev} - y_{thresh}$
 j_{max} // averaging window size
 m // number of consecutive reversals in a filter
// parameter before convergence declared

```

1:  $y \leftarrow \text{abs}(x_1 - x_0) + k_{\text{corr}}$  // Calculate the current eye size.
2:
3: if  $y < y_{\text{min}}$  or  $x_1, x_0$  undefined // If a valid eye opening has not been found,
4: INITIAL SEARCH: // run initial search stage:
5:      $\text{allConverged} \leftarrow \text{false}$  // Reset the allConverged flag.
6:
7:     if  $h_i = h_{i,\text{max}}$  // If current filter parameter is saturated,
8:          $h_i \leftarrow h_{i,\text{min}}$  // reset it and
9:          $i \leftarrow i + 1$  (looping back to 1) // go to the next filter parameter.
10:    end
11:
12:     $h_i \leftarrow h_i + h_{i,\text{step}}$  // Increment current filter parameter.
13: GUIDED ADAPTATION/CONVERGED MONITORING:
14: else // If a valid eye has been found,
15:      $y_{\text{accum}} \leftarrow y_{\text{accum}} + y$  // add current eye size to accumulator and
16:      $j \leftarrow j + 1$  // increment averaging counter.
17:
18:     if  $j = j_{\text{max}}$  // If averaging complete,
19:          $z \leftarrow y_{\text{accum}}/j_{\text{max}}$  // calculate the current average eye size.
20:     else if  $y < z_{\text{tgt}} - y_{\text{thresh}}$  // Otherwise, if current eye size falls below
21:         // minimum threshold,
22:          $y_{\text{accum}} \leftarrow 0$  // reset accumulator,
23:          $z_{\text{tgt}} \leftarrow 0$  // reset target eye size and
24:          $\text{allConverged} \leftarrow \text{false}$  // reset the allConverged flag.
25:     end
26:
27:     if  $\text{allConverged} = \text{false}$  // If filter not declared converged,
28: GUIDED ADAPTATION: // run guided adaptation.
29:     if  $z < z_{\text{tgt}}$  // If current average eye size less than target,
30:          $h_{i,\text{dir}} \leftarrow -h_{i,\text{dir}}$  // reverse current parameter step direction and
31:          $h_i \leftarrow h_i + h_{i,\text{dir}} \cdot h_{i,\text{step}}$  // step it, so it reverts to its previous value.
32:     else // Otherwise,
33:          $z_{\text{tgt}} \leftarrow z$  // current average is better, so it becomes the
34:     end // new target.
35:
36:      $H_{\text{conv}} \leftarrow (\text{false} \dots \text{false})$  // Setup a vector to hold parameter
37:     // convergence state.
38:     for  $k = 1$  to  $n$  // For each parameter,
39:         if  $h_k$  has reversed step direction more than  $m$  times consecutively
40:              $h_{k,\text{conv}} \leftarrow \text{true}$ 
41:         end
42:     end

```

```

43:     if  $H_{conv} = (\text{true} \dots \text{true})$            // If all individual parameters are converged,
44:          $allConverged \leftarrow \text{true}$          // set the allConverged flag.
45:     else                                         // Otherwise,
46:          $i \leftarrow i + 1$  (looping back to 1) // go to the next filter parameter and
47:          $h_i \leftarrow h_i + h_{i,dir} \cdot h_{i,step}$  // step it in the specified direction.
48:     end
49: end
50: end

```

Although the mismatch counter in the ping-pong CDR is designed to produce reasonably consistent estimates of eye size and location (see section 3.1.2), over finite intervals this estimate is nevertheless sequence-dependent. Therefore, the eye size estimates are averaged over j_{max} results before they are used to adapt the equalizer. The equalization adaptation itself proceeds in binary or bang-bang (up/down) fashion, changing one filter parameter at a time. This makes the adaptation process somewhat slow, especially if there are many parameters to adapt. Typical wireline communications channels (whether on-chip or chip-to-chip) do not change much over time, however, so this is of little concern.

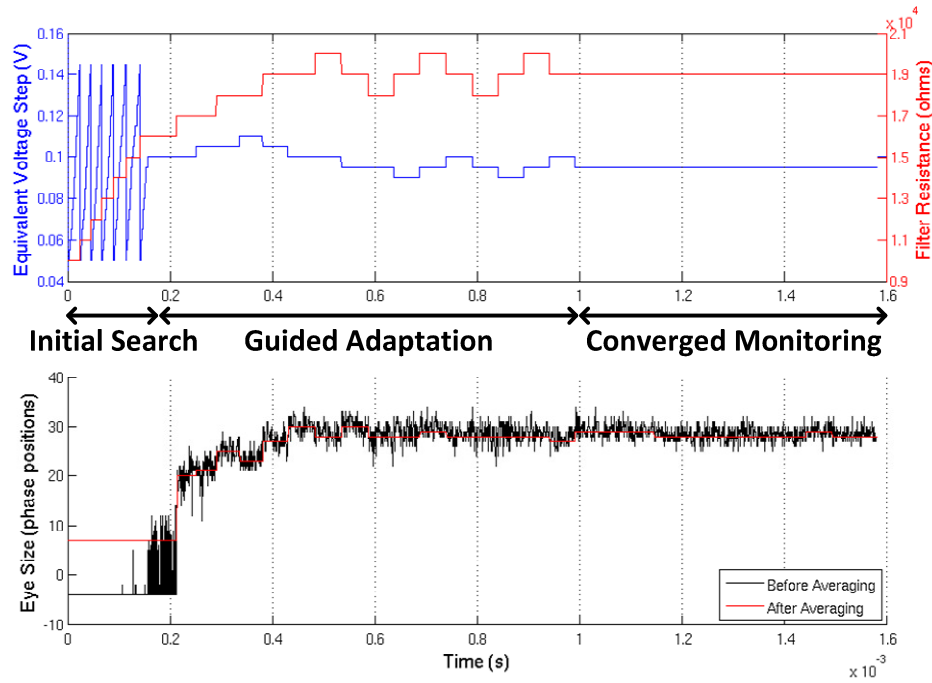


Figure 3.34: Simulation of ping-pong CDR-based adaptive equalizer, showing stages of operation.

The system described was applied to a DFE with first-order IIR feedback (Figure 3.33(b)) and simulated in MATLAB, using PRBS-31 data at 5 Gb/s over the RC-limited on-chip channel described in Figure 3.31. Figure 3.34 shows a sample simulation run, starting with no eye opening and running through the initial search stage until an eye opening is detected. Eye opening information is then used to run the guided adaptation until all filter parameters (voltage step and resistance in this case) have met the convergence condition specified by m , whereupon the converged monitoring phase is entered. During converged monitoring, tap weights are frozen and the system monitors the detected eye size; if the eye size ever falls below $z_{tgt} - y_{thresh}$, the system returns to the guided adaptation mode. The simulation displayed $BER < 10^{-6}$, limited only by the time necessary to simulate a large number of bits. For the sake of comparison, a conventional DFE with 7-tap FIR-based feedback and SS-LMS adaptation run at 3.25 Gb/s over the same channel produced roughly equivalent voltage margins at the sample point (Figure 3.35).

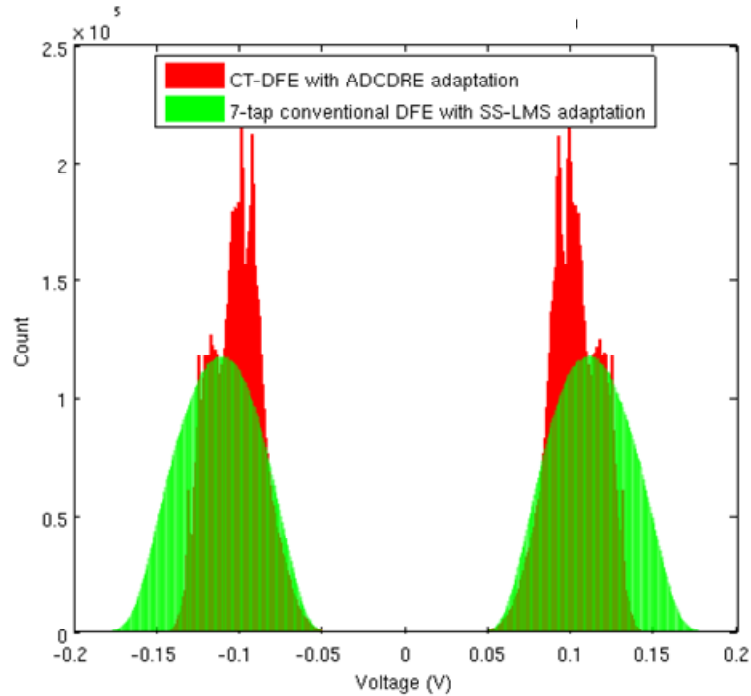


Figure 3.35: Histograms of received signal level at sample point, comparing continuous-time IIR DFE with ping-pong CDR-based adaptation @ 5 Gb/s (red) against 7-tap FIR DFE with SS-LMS adaptation @ 3.25 Gb/s (green)

3.6 Summary

Per pin-phase alignment is an important part of high-speed chip-to-chip interconnect. However, the power and area overheads for traditional DLL or PLL-based solutions are prohibitive, and simpler alternatives, such as open-loop delay lines, cannot realize the infinite delay range required in large, noisy digital systems. The ping-pong CDR overcomes these limitations by pursuing a statistical approach to eye detection, using the eye information to optimize the timing margins available to the sampling clock. This approach is readily implemented in a highly digital manner, and lends itself well to standard synthesis and place-and-route workflows, reducing design time and easing integration in highly-scaled CMOS process technologies.

Further efficiency can be achieved by sharing the ping-pong CDR over multiple data pins. To swap functions between data and search, it implements generalized sampling and re-timing paths; instead of swapping data and search for one data pin, these paths can be swapped across multiple pins. Sharing the CDR in this fashion reduces clocking and control logic overheads, and careful selection of the order in which the data pins are calibrated limits the increase in multiplexer and signal routing complexity. Although sharing inevitably reduces CDR bandwidth, a 3-pin implementation in 90 nm CMOS showed power savings of about 32%.

Looking beyond chip-to-chip interconnect, the trend towards integrating larger numbers of microprocessor cores on a single die has resulted in an increase in the number and length of long on-chip wires. Signal integrity considerations for communication over these wires are very similar to those in chip-to-chip communication, and timing recovery and adaptive equalization are an integral part of on-chip signalling in the many-core era. The ping-pong CDR is a natural fit in this environment, since it is easy to integrate and share. Importantly, the eye-monitor data that it collects can be used to adapt unconventional DFE types (such as IIR-DFE), which are particularly well-suited to the long-tail ISI produced by on-chip wires.

Chapter 4: Proximity Communication

The notion of communicating via electrostatic coupling or magnetic induction found its genesis early in the development of wireless communication technology. Initial attempts predate Hertz's discovery of electromagnetic waves (as predicted by Maxwell) in 1887; for example, in 1885 Edison filed a patent describing a wireless telegraphy system using capacitive coupling between plates ('condensing-surfaces') suspended in the masts of ships, from tall poles or from balloons, with the intervening air acting as the dielectric [68]. Although this (rather ambitious) scheme was never tested, the same idea formed the basis of a system designed for communications with a moving train, involving capacitive coupling between a plate spanning the entire length of a rail car and existing telegraph wires next to the track [69], [70]. Although successfully demonstrated and actually implemented commercially, lack of demand caused the end of service; apparently business travellers at the time preferred 'to be free from telegrams of all sorts "while on the wing"' [70] – a stark contrast to today's demand for pervasive mobile communication.

Experiments were also attempted using inductive coupling. In 1892, Charles A. Stevenson, of the Northern Lighthouse Board in Edinburgh and naturally concerned with the problems of navigation off-shore in inclement weather, proposed laying a underwater cable extending from a shore-based transmitter, over which a ship could pass. The ship itself would have a wire extended along its length, terminated in coils at each end. The coils would pick up the magnetic field induced by the submerged cable as the ship crossed over, allowing the ship to determine its heading and/or location. He later experimented with both coil-to-coil and parallel-wire inductive coupling systems, successfully communicating over a distance of 870 yards. An agreement was reached to install a similar system for the purpose of communicating to a lighthouse on Muckle Flugga, and island off the coast Scotland; financial difficulties eventually ended the project [70].

Perhaps the most impressive of these early attempts, both in terms of the distance of communication as well as the size of the coupled elements used, was made by Sir William H. Preece, a Welsh engineer. In 1894, he constructed an inductively-coupled telegraph link, spanning a 4 mile distance across the Kilbrannan Sound in Scotland. The parallel wires used were between 2 and 4 miles long, and Preece ultimately concluded that the length of the wires needed to be at least equal the distance between them in order to facilitate reliable communication, making the system impractical for most purposes. It did, however, find a few notable uses, the first of which occurred when the regular cable telegraph between the Isle of Mull and Oban (again in Scotland) broke down on March 30, 1895. A 1.5 mile wire was stretched along the ground in the nearby Morven peninsula, and communication was successfully established by coupling to an existing telephone wire on the Isle of Mull, allowing the continuance of telegraph operation until the regular cable was repaired a week later [70].

The advent of radio-frequency (in the parlance of the time, ‘Hertzian wave’) communication, which could achieve much superior range with more reasonably-sized transmitters/receivers, rendered attempts at signalling via capacitive or inductive coupling obsolete. As a result, the field languished for almost a century until the advent of IC design and CMOS scaling sparked a rapid rise in the complexity of electronic systems, and a corresponding decrease in their physical form-factor. In this context, the idea of using capacitive or inductive coupling resurfaced, no longer for wireless communication over long distances, but instead to enable short-distance wireless communication - first as a way to increase interconnect density between chip and package [71], and later to facilitate chip-to-chip communication for 3D integration [72], [9], [73]. The short-distance nature of such interconnect led to it being named ‘Proximity Communication’, and it has assumed three primary forms: the capacitive and inductive types familiar from earlier experiments, and an optical type enabled by recent developments in silicon photonics [74].

By focusing attention on only the short (microns-to-millimetres) distances required for in-package communication, more recent attempts at capacitively or inductively coupled communication avoid the range and size issues faced by earlier investigations attempting to span miles-long distances. Indeed, because the coupled elements (whether plates of a capacitor or spirals of an inductor) can be realized in the existing metal stack of the IC (therefore requiring no

special new processing steps), and no physical connection need be made (avoiding yield issues inherent in the bump-bonding process, and eliminating the requirement for ESD structures, which consume valuable die area and add parasitic load), proximity communication can achieve densities in excess of traditional wired connections in a cost-competitive way.

4.1 Capacitive Proximity Interconnect

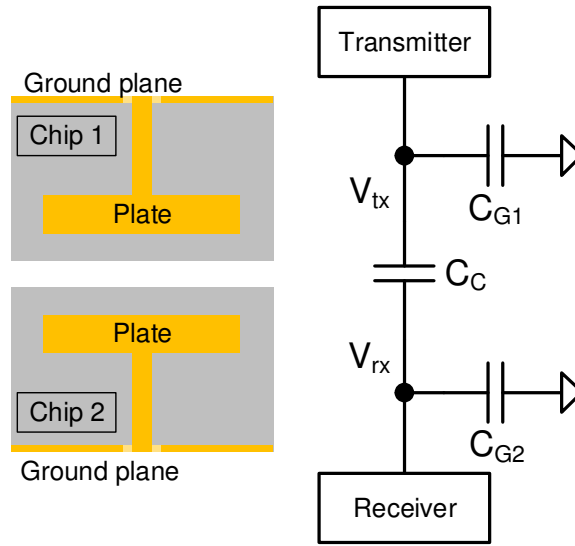


Figure 4.1: Basic structure of capacitive proximity interconnect, with equivalent circuit

At its heart, capacitive proximity interconnect uses small metal plates in the uppermost (typically, pad-level) metal layer of two ICs. When these ICs are brought close together and face-to-face, the plates in both chips form capacitors that allow information to be sent from one side to the other (Figure 4.1). The transfer function, from transmitter to receiver, is a straightforward capacitive voltage divider:

$$H(s) = \frac{V_{rx}}{V_{tx}} = \frac{C_C}{C_C + C_{G2} + C_{rx}} \quad (4.1)$$

where C_{rx} is the input capacitance of the receiver.

The area of the plates (A) to be used is an important design parameter. Assuming that the plates are relatively large compared to the distance between them (d_C), parallel-plate fields will essentially dominate the coupling capacitance (C_C). Since the distance, d_G , between plate and

ground plane (on some lower-level metal layer) is likely to be small as well, the same can be said for C_{G2} . Therefore, the two capacitances can be expressed as:

$$C_C \approx \frac{A \cdot \epsilon_{eff} \cdot \epsilon_0}{d_C}, \quad C_{G2} \approx \frac{A \cdot \epsilon_{ox} \cdot \epsilon_0}{d_G} \quad (4.2)$$

where ϵ_{eff} is the effective dielectric constant between the two plates, and ϵ_{ox} is the dielectric constant of the oxide (or, in more advanced processes, low-k dielectric) between the metal layers.

Substituting these relationships into (4.1) yields:

$$H(s) \approx \frac{\frac{A \cdot \epsilon_{eff} \cdot \epsilon_0}{d_C}}{\frac{A \cdot \epsilon_{eff} \cdot \epsilon_0}{d_C} + \frac{A \cdot \epsilon_{ox} \cdot \epsilon_0}{d_G} + C_{rx}} \quad (4.3)$$

Since C_{rx} is independent of A , gain from transmitter to receiver will increase with A , albeit with diminishing returns, until $C_C + C_{G2} \gg C_{rx}$. At this point, any further enlargement of plate area serves to increase coupling capacitance and parasitic capacitance equally and maximum gain has been achieved:

$$H(s) \approx \frac{\frac{\epsilon_{eff}}{d_C}}{\frac{\epsilon_{eff}}{d_C} + \frac{\epsilon_{ox}}{d_G}} = \frac{\epsilon_{eff} \cdot d_G}{\epsilon_{eff} \cdot d_G + \epsilon_{ox} \cdot d_C} \quad (4.4)$$

In principle, then, the maximum useful plate size occurs when

$$A \gg \frac{C_{rx} \cdot d_C \cdot d_G}{\epsilon_0(\epsilon_{eff} \cdot d_G + \epsilon_{ox} \cdot d_C)} \quad (4.5)$$

In practice, determining the optimum plate size is complicated by the fact that neither the dielectric between the two plates nor that between the metal layers is homogenous, the presence of adjacent plates in the array, crosstalk considerations, the effect of plate size on wiring parasitics, the desired communication density and other concerns.

Since the value of a parallel-plate capacitor is inversely proportional to the distance between the plates (i.e. $C_C \propto 1/d_C$), capacitive proximity interconnect tends to be very sensitive to misalignment between the two chips. As a result, alignment needs to be very well controlled ($< 10 \mu\text{m}$ offset in any direction) unless steps are taken to adapt the array to non-ideal alignment [10]. Such adaptation naturally requires a means to sense the alignment in the first place, and approaches to tackling this problem are described in sub-section 4.1.2.

4.1.1 Transmitter and receiver design

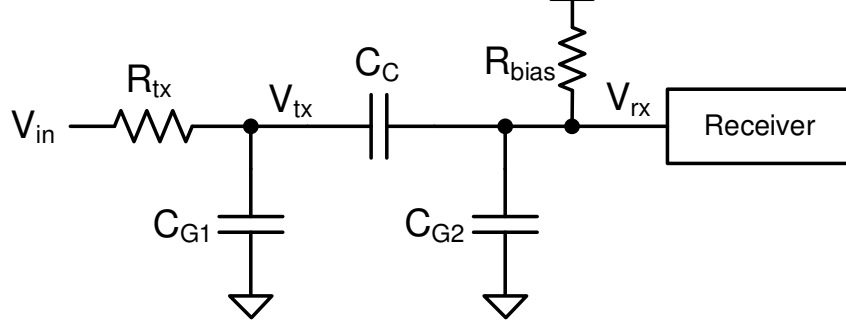


Figure 4.2: Capacitive link, showing transmit driver resistance and receiver bias resistance

The transfer function of the capacitive link (4.1) is ideally frequency-independent. This suggests that relatively simple transmitter and receiver circuits should be possible, with no need for complex pre-emphasis or equalization schemes. However, the output resistance of the transmitter and the resistance of the receiver's input bias circuitry create a band-pass filter (Figure 4.2), modifying the overall link transfer function to:

$$H(s) = \frac{V_{rx}}{V_{in}} = \frac{sC_c}{(1 + sR_{bias}(C_{G2} + C_{rx} + C_c))(1 + sR_{tx}(C_{G1} + C_{tx} + C_c)) - s^2C_c^2R_{tx}} \quad (4.6)$$

Since the coupling capacitance (C_c) is generally much lower than either parasitic capacitance ($C_{G2} + C_{rx}$ or $C_{G1} + C_{tx}$), the transfer function can be simplified:

$$H(s) \approx \frac{sC_c}{(1 + sR_{bias}(C_{G2} + C_{rx}))(1 + sR_{tx}(C_{G1} + C_{tx}))} \quad (4.7)$$

In general, $C_{G2} + C_{rx} \approx C_{G1} + C_{tx}$ and $R_{bias} > R_{tx}$, so the passband exists between

$$\omega_l \approx \frac{1}{R_{bias}(C_{G2} + C_{rx})}, \quad \omega_h \approx \frac{1}{R_{tx}(C_{G1} + C_{tx})} \quad (4.8)$$

R_{tx} and the parasitic capacitances should be kept as small as possible in order to avoid the need for equalization. This is not normally a challenge for data rates up to 10 Gb/s; for example, if a simple inverter is used as the transmitter, R_{tx} (in a typical 65 nm CMOS process) can easily be made $< 500 \, \Omega$. Plate sizes range from $20 \times 20 \, \mu\text{m}$ [73] to $35 \times 35 \, \mu\text{m}$ [9], for parasitic capacitances $< 50 \, \text{fF}$, so $f_h > 6.4 \, \text{GHz}$.

On the other hand, achieving a high R_{bias} , such that the passband extends low enough to support the transmission of baseband signals, is somewhat more challenging. A simple resistor will not do, since there is rarely enough space on-chip to implement resistances in the 10's to 100's of $M\Omega$ required to push the passband below 1 MHz; in any case, such a large resistor would add a significant amount of parasitic capacitance at the receiver, reducing the passband gain substantially.

One way to deal with this problem is to simply accept that a certain amount of high-pass filtering is going to happen, and use a variable-threshold input comparator at the receiver to detect the voltage pulses that occur when there is a transition in the transmitted signal. For example, the design in [9] used a set of weak feedback switches (M1 and M2 in Figure 4.3) to drive the receiver plate to a high (V_h) or low (V_l) bias voltage after each transition in the input signal. The output of the inverter-based input comparator would therefore stay stable until a subsequent transition in the input signal caused a voltage pulse large enough to cross its threshold voltage. The key challenge to this design is generating V_h and V_l reliably, at values far away enough from the nominal comparator threshold voltage to tolerate process variation and provide some level of noise margin, while still setting them close enough to the threshold so that even very small pulses can be detected, in the event that coupling between the two plates is poor.

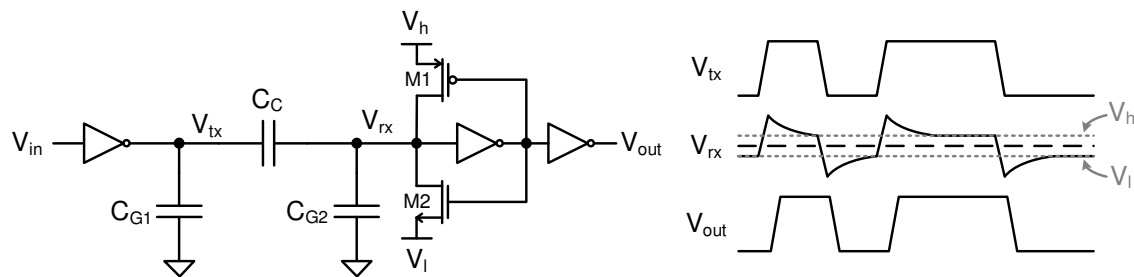


Figure 4.3: Variable-threshold comparator used as a receiver in a capacitive proximity interconnect, with waveforms showing principle of operation

An important disadvantage of this approach is that the feedback switches will contend with the voltage pulse, so weakening it. The solution to this problem is to take a precharge-and-evaluate approach. During the precharge phase, the bias point is defined by shorting the receive plate to a known voltage. During the evaluate phase, the plate is allowed to float, and responds

to transitions in the input. This response can be amplified and then re-timed using a flip-flop or senseamp (Figure 4.4). This method that has been applied both to synchronous (i.e. clocked) [73], [75] as well as asynchronous [76] capacitive proximity interconnect. In the synchronous case, its primary disadvantage is that half the clock period (when it is high, say) is expended in the precharge phase, reducing the available timing margin.

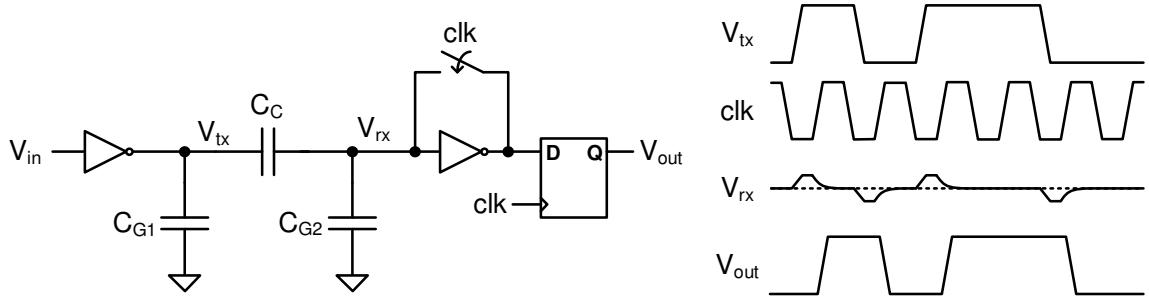


Figure 4.4: Example precharge-and-evaluate input stage (adapted from [75]). During precharge (clock is high), bias point is set by shorting the input and output of the input inverter stage, causing it to enter its metastable state. The D flip-flop evaluates the current bit at the rising edge of the clock

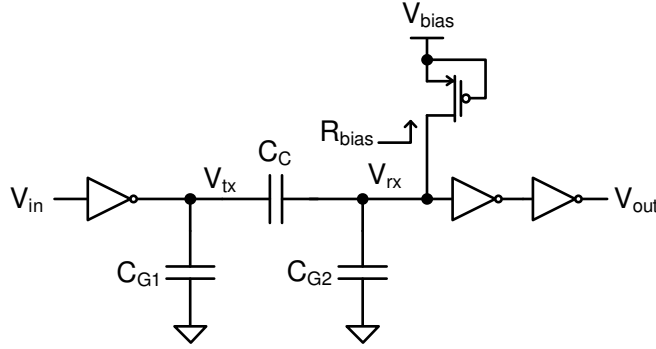


Figure 4.5: High R_{bias} achieved using leakage device

The most direct approach avoids the timing requirements of the precharge-and-evaluate method by using a leakage device to define the DC bias [10] (Figure 4.5). By taking advantage of the small sub-threshold current in deep sub-micron transistors, a sufficiently high (if poorly defined) effective R_{bias} can be achieved. However, precisely because R_{bias} is so large, the receiver input is only very weakly tied to its bias voltage. As a result, the transmitted bit sequence needs to be close to (but not exactly) DC balanced to prevent large drifts in the bias point.

4.1.2 Sensing and adapting to misalignment

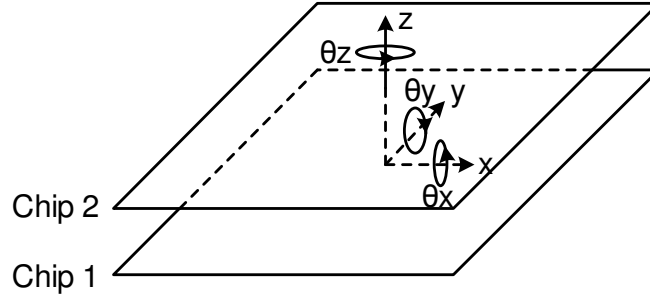


Figure 4.6: Chip-to-chip alignment can be described in terms of 6 axes: 3 linear (x , y and z) and 3 angular (θ_x , θ_y , θ_z)

As described above, capacitive coupling is very sensitive to the alignment between the plates in the two chips. Although manufacturing techniques exist to ensure the vertical (z -axis in Figure 4.6) separation between the two chips is small [77], vibration or thermal expansion may cause the in-plane (x - and y -axis) alignment between the two chips to vary over time. A means of sensing these changes in alignment and adapting the capacitive link to them are therefore important.

Alignment sensors come in two different basic types – those that use a dedicated set of plates/bars, and those that use the existing communication array to sense alignment. Dedicated sensing arrays have obvious advantages: they are not restricted to the plate geometries and sizes optimal for communication and can use specialized sensing circuitry without fear of imposing an excessive load on the communications link. As a result, they can achieve better resolution than sensors embedded in the communication array. However, the structures that result can be quite large (as much as several times the size of the communication array itself [78]), reducing communication density versus using the same set of plates for communication and alignment. Sensors sharing plates with the communication array have the additional advantage of measuring the link quality directly, which is particularly useful in designs where limited computational power is available and calculating link quality based on the output of dedicated alignment sensors is prohibitively expensive.

Naturally, dedicated alignment sensors have assumed many different shapes. One approach uses sets of bars in the two chips, each chip having a different number of bars spread over the same distance [9], [10], [78]. The result is a vernier system whose accuracy is defined by the

difference in pitch between the two sets of bars. Measurements are taken by stimulating a transmitting set of bars with positive or negative edges (alternating between bars), amplifying the received waveform at the opposing set of bars and capturing the result using a flip-flop. Depending on which transmitting bar each receiver happens to be closer to, the captured result will either be a high or low, and the overall bit pattern will indicate the relative alignment between the two chips. This vernier bar system has the advantage of high accuracy (as good as 1 μm has been achieved [78]), and is easy to use since its output is inherently digital. On its own, however, it is incapable of measuring z-axis separation, and therefore covers only three (x, y and θ_z) axes out of the six. Additionally, due to the high-pass nature of the channel (a feedback resistor is used to bias the receiver amplifier), the clock edge used to capture the result (Rx Edge) must be well-timed with respect to the stimulus edge at the transmitter (Tx Edge).

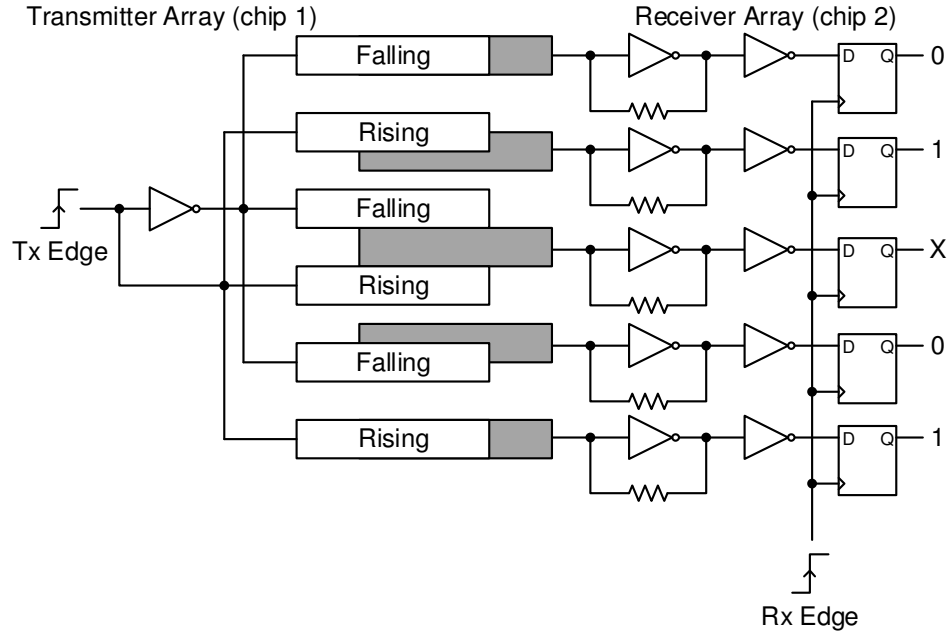


Figure 4.7: Vernier bar-based alignment sensor [9]. Output of flip-flop depends on the most strongly coupled transmitter bar, and can be unknown ('X') if two adjacent transmitter bars are equally-well coupled

The vernier bar arrangement can be modified to sense z-axis (and, by comparing results across the array, θ_x and θ_y) alignment by adding a rectifier to the sensing circuitry [78]. When a clock signal is feed to the transmitting bar, the rectifier captures the amplitude arriving at the

receiver, from which the amount of coupling between the two can be inferred. Comparisons with field-solver simulations can then be used to determine the z-axis separation between the two chips.

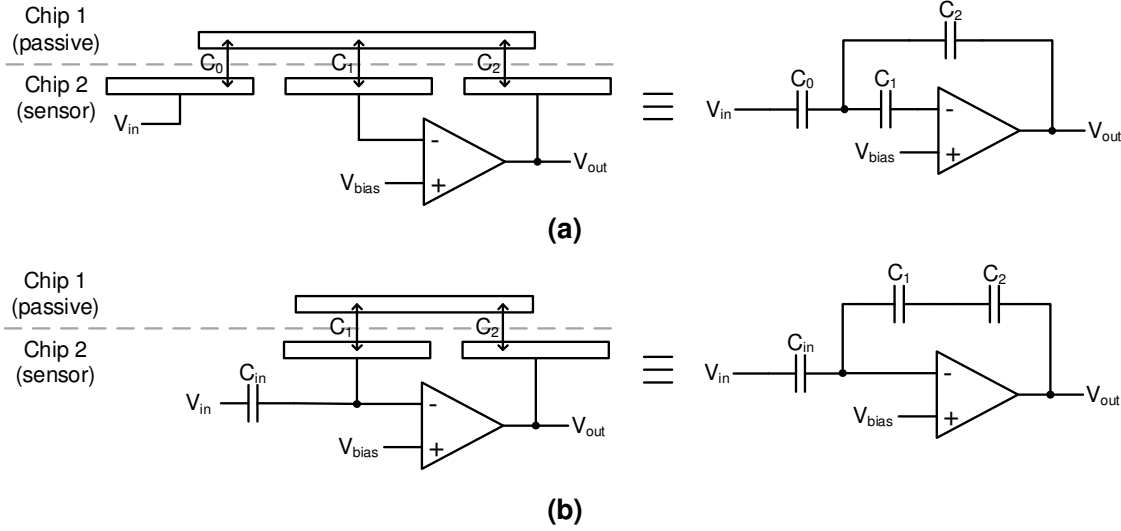


Figure 4.8: Multi-plate alignment sensor with passive target chip [79], for (a) in-plane (x- and y-axis) alignment and (b) vertical (z-axis) alignment, showing equivalent circuits

The amplitude of a coupled signal can also provide alignment information in the case where one of the chips is unpowered and therefore incapable of driving a stimulus. A multi-plate (2 plates for z-axis sensing, and 3 for x- or y-axis) design is used [79] (Figure 4.8), where one of the plates is used to couple in a stimulus from the powered sensor chip to the passive target chip. This stimulus is then coupled back onto the sensor chip using the remaining plate(s), so that alignment can be measured. A limitation of this method is the requirement for sufficient coupling to all three (or two) plates in order for the sensor to function correctly, so these plates need to be big enough to accommodate any anticipated misalignment. Both this method, as well as the z-axis extension to the vernier bar method described previously, produce an analog voltage as output. Consequently, a further analog-to-digital converter (ADC) is required if the results are to be used for array adaptation.

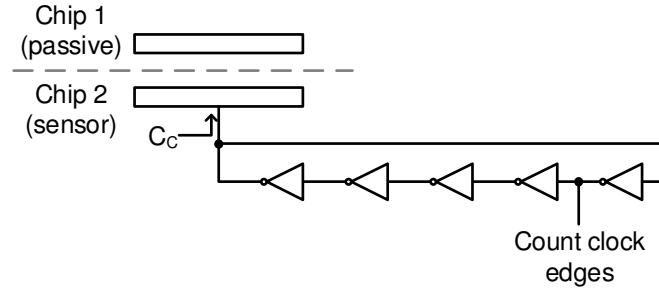


Figure 4.9: Ring oscillator-based alignment sensor [80]

Embedding an alignment sensor into the communication array is appealing where available power and space are limited. For example, an embedded alignment sensor has been used in a capacitively coupled power delivery and communications system to ultra-low-power sensor chips [80]. In this case, a ring oscillator was associated with each communications/sensor plate, with the plate acting as a load on one stage of the ring oscillator (Figure 4.9). Since the plate coupling capacitance (C_C) increases when it is better-coupled to a plate in the opposing chip, and the ring oscillator output frequency is dependent on the amount of capacitance seen by the plate, alignment can be determined by counting the number of ring oscillator periods over a specified time interval. By using delay, rather than voltage, as the intermediate quantity between alignment and digital output, this scheme is able to use a much simpler, more compact and lower-power ADC. An additional advantage is its ability to generate (in principle) infinite resolution by simply allowing the ring oscillator period counter to run for longer intervals, noting the number of times it rolls over. In practical terms, resolution is limited by the available waiting time, as well as power supply, crosstalk or other noise. Indeed, care has to be taken to ensure that such noise does not cause ring oscillators from different plates to injection-lock to each other, degrading results. By attempting to sense an undriven capacitance, this scheme is limited in its ability to distinguish between capacitance that results from coupling between plates and that from the ground-plane under the plates. Furthermore, the need to embed a whole ring-oscillator and set of counters (although certainly superior to a voltage-based system) does ultimately limit the achievable sensor plate density (e.g. to a pitch of about $55\text{ }\mu\text{m}$ in [80]).

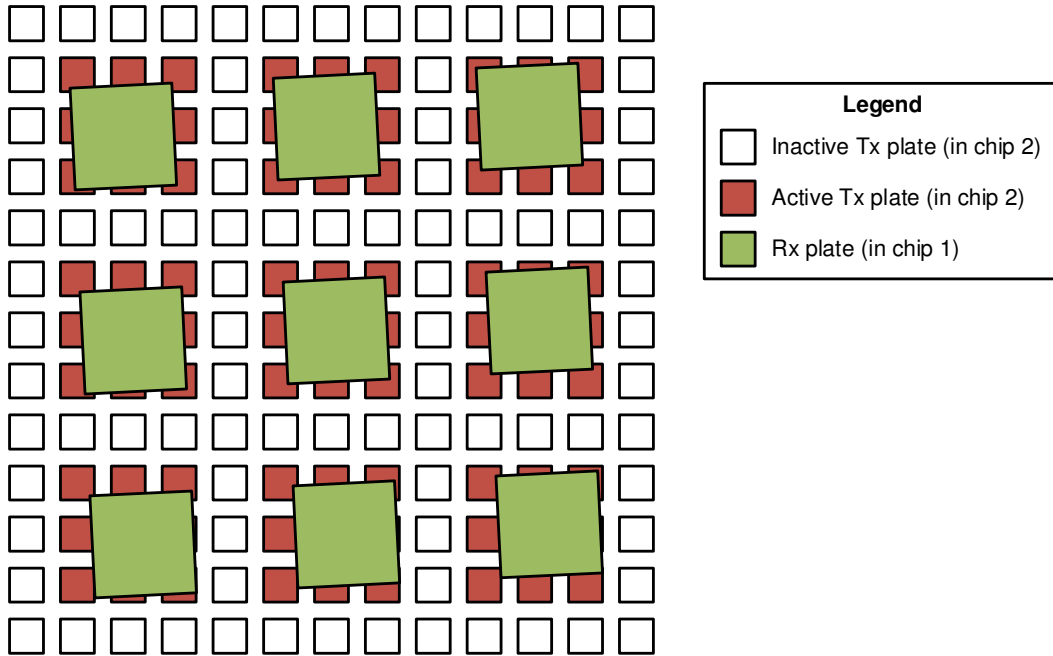


Figure 4.10: Adaptation of transmitter array to receiver array alignment [81]

Once the alignment has been determined, the transceiver arrays need to be adapted to optimize communication for that alignment. The basic principle is quite straightforward – split one of either the transmitter or receiver arrays up into smaller plates, then combine those plates together to maximize overlap between transmitter and receiver (Figure 4.10). Since combining the plates together involves switches and wiring, all of which add extra parasitic capacitance, it makes more sense to split up the transmitter plates (where the only penalty is extra power consumption) than the receiver plates (where the penalty is a reduction in gain from transmitter to receiver), and this is indeed the approach taken by most high-data-rate implementations (e.g. [81], [10]). Other considerations may force a different approach, however – for example, if the same array is used both to deliver power to, as well as receive data from, a coupled chip [80]. Another important consideration is the range of plates over which a particular data pin should be steerable. Greater flexibility requires more complex wiring and a larger number of multiplexers, rapidly complicating design. As such, the desired misalignment tolerance needs to be carefully determined before designing the switch fabric.

4.2 Inductive Proximity Interconnect

The primary limitation of capacitively-coupled proximity interconnect is range – as described above, typical implementations call for alignments better than 10 μm in any directions, and take considerable pains to sense any misalignment and correct for it as well as possible. For face-to-face communication between two stacked chips, this is a surmountable obstacle. However, for multi-chip stacking applications, or scenarios where one of the chips has to face away from the other (e.g. if the bottom chip is flip-chip bonded and therefore face-down), the longer-range inductively-coupled links become an appealing alternative [82].

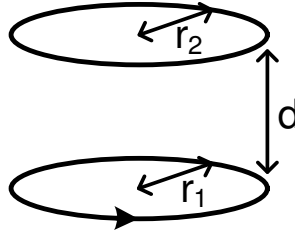


Figure 4.11: Circular wire loops for calculation of mutual inductance

Whereas capacitively-coupled proximity interconnect relies fundamentally on the capacitance achieved between two parallel plates, which scales inversely with plate separation and is therefore limited in range (4.2), inductive coupling uses the mutual inductance achieved between two coils of wire. This decays less quickly with distance than (mutual) capacitance; as a simple example, consider the mutual inductance between two single loops of wire (Figure 4.11). From [83]:

$$M = \mu_0 \sqrt{r_1 r_2} \left[\left(\frac{2}{b} - b \right) K(b) - \frac{2}{b} E(b) \right] \quad (4.9)$$

Where $K(b)$ and $E(b)$ are complete elliptic integrals of the first and second kind, respectively, and

$$b = \sqrt{\frac{4r_1 r_2}{(r_1 + r_2)^2 + d^2}} \quad (4.10)$$

The rate of decay in mutual inductance between two loops is compared with that of capacitance between two parallel plates (excluding fringing fields) in Figure 4.12; it is readily apparent from this comparison that the mutual inductance decays much more slowly. Note that this simple

comparison excludes the effects of multiple coils in the coupled inductor, which would increase the mutual inductance and make the comparison even starker.

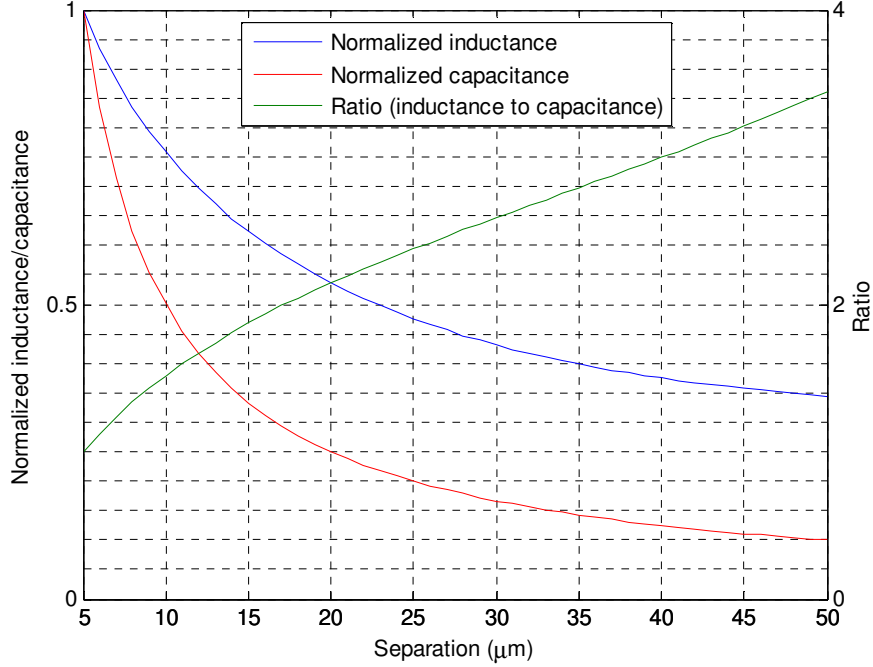


Figure 4.12: Comparison of mutual inductance between two single-loop wires with $r_1 = r_2 = 50 \mu\text{m}$ and capacitance between two $50 \times 50 \mu\text{m}$ square parallel plates in silicon dioxide ($\epsilon_r = 3.9$), over different amounts of separation

An alternative way to characterize the mutual inductance is via the coupling coefficient, k , which is useful since it measures the mutual inductance as a proportion of the self-inductance. In general,

$$k = \frac{M}{\sqrt{L_1 L_2}} \quad (4.11)$$

where L_1 and L_2 are the self-inductances of the two coils. The self-inductance of a circular wire loop with one turn is given by [83]

$$L \approx \mu_0 r \left(\ln \frac{8r}{a} - 2 \right) \quad (4.12)$$

where r is the radius of the loop and a is the radius of the wire. If $r_1 = r_2 = r$ (hence, $L_1 = L_2$), (4.11) becomes

$$k = \frac{M}{\mu_0 r \left(\ln \frac{8r}{a} - 2 \right)} \quad (4.13)$$

Figure 4.13 plots the coupling coefficient for the same loops as those used for Figure 4.12. Since the self-inductance of the loops does not change with separation, the shape of the curve is the same as that for the normalized mutual inductance.

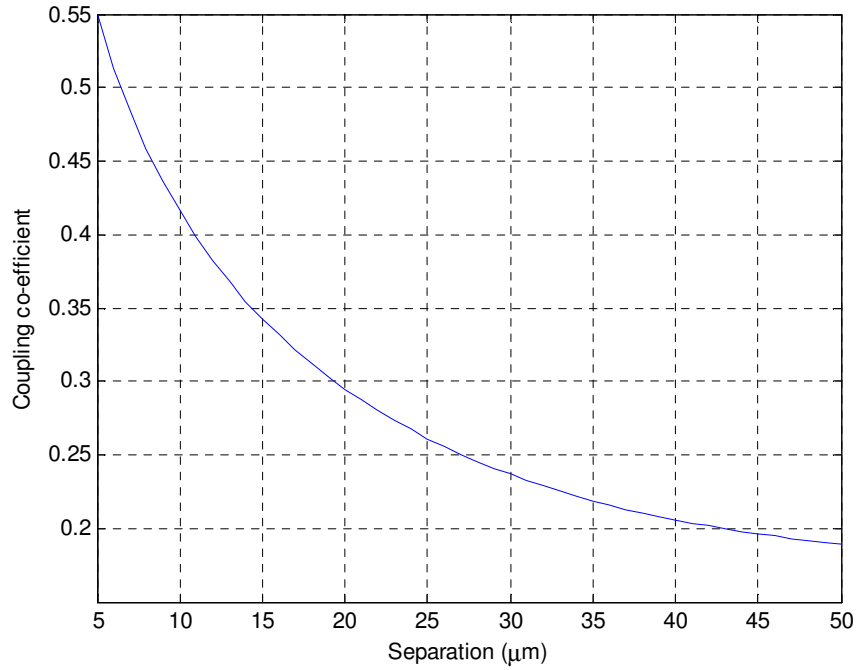


Figure 4.13: Coupling co-efficient between two single-loop wires with $r_1 = r_2 = 50 \mu\text{m}$, over different amounts of separation

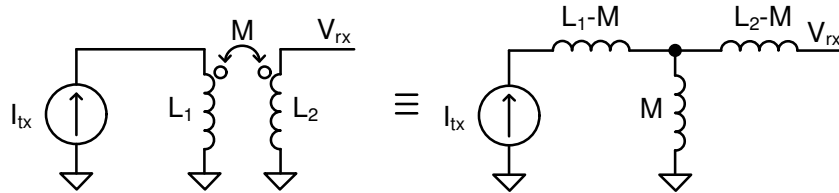


Figure 4.14: Ideal inductively-coupled link, with equivalent circuit

The transfer characteristics of an inductively-coupled proximity link can be derived in much the same way as those for a capacitively-coupled one, with the exception that the input is better

modelled as a current rather than a voltage. Taking first the simple case, where the inductors are ideal and have no loss or associated parasitic capacitance (Figure 4.14):

$$H(s) = \frac{V_{rx}}{I_{tx}} = s \cdot M = s \cdot k \sqrt{L_1 L_2} \quad (4.14)$$

suggesting that the link acts as a differentiator, and has a transfer characteristic dependent only on the mutual inductance.

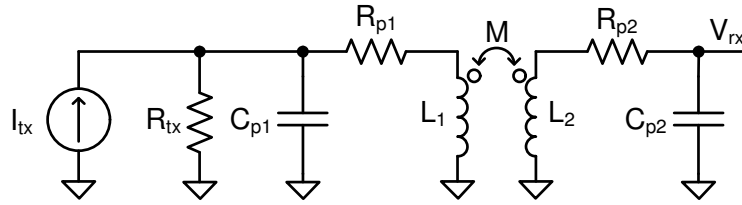


Figure 4.15: Inductively-coupled link with parasitics.

The introduction of inductor parasitics change this picture considerably; Figure 4.15 provides an equivalent circuit model of these parasitics, where R_{p1} and R_{p2} are coil resistances, C_{p1} and C_{p2} include both the coil parasitic capacitance and the transmitter output capacitance or receiver input capacitance, respectively, and R_{tx} is the output resistance of the transmitter. Note that the inductor parasitics have been lumped into single elements, a reasonable approximation since the inductors used have diameters $< 100 \mu\text{m}$ and the wavelength in silicon dioxide is approximately 7.7 mm @ 10 GHz . A full analytical expression for the transfer characteristic of this circuit is too complex to provide useful insight into its operation. Instead, it is helpful to observe that the input impedance of the receiver is very high (it is primarily a small capacitance) for the $< 10 \text{ GHz}$ frequencies being considered. As a result, the currents induced in the receiver coil are very small, and there is only a small amount of feedback current induced in transmitter coil [82]. It is therefore possible to approximate the coupled coils as a unilateral system, yielding the equivalent circuit in Figure 4.16.

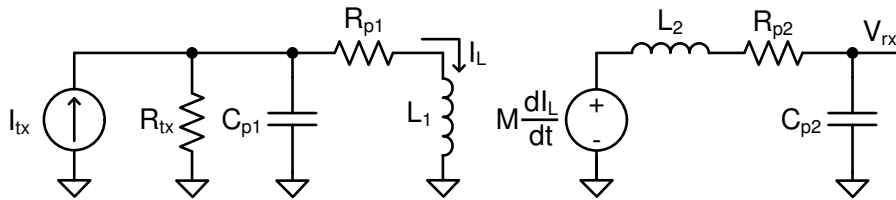


Figure 4.16: Simplified inductively-coupled link model, using a unilateral coupled inductor

This has transfer function:

$$H(s) = \frac{V_{rx}}{I_{tx}} = sM \cdot \frac{R_{tx}}{s^2 R_{tx} C_{p1} L_1 + s(C_{p1} R_{tx} R_{p1}) + R_{tx} + R_{p1}} \cdot \frac{1}{s^2 C_{p2} L_2 + s C_{p2} R_{p2} + 1} \quad (4.15)$$

The first term is clearly the current derivative response of the ideal link described in (4.14), while the second and third terms are second-order LPFs created by the transmitter and receiver parasitics, respectively. The resonant frequencies for these filters are:

$$\omega_{0,tx} = \sqrt{\frac{R_{tx} + R_{p1}}{R_{tx} C_{p1} L_1}} \approx \frac{1}{\sqrt{C_{p1} L_1}}, \quad \omega_{0,rx} = \frac{1}{\sqrt{C_{p2} L_2}} \quad (4.16)$$

The approximation in $\omega_{0,tx}$ can be made since the current source used to drive the link generally has $R_{tx} \gg R_{p1}$. The bandwidths are:

$$\omega_{b,tx} = \frac{R_{p1}}{L_1}, \quad \omega_{b,rx} = \frac{R_{p2}}{L_2} \quad (4.17)$$

The peak in the frequency response introduced by the second-order LPFs is detrimental to the performance of the link, since the resulting ringing creates ISI. Although the ringing can be reduced by adding de-Q resistors in series with both the receiver and transmitter coils, this has not been done in existing designs. Instead, they have simply limited data rates such that [8]

$$T_b > \frac{4}{\pi f_0} \quad (4.18)$$

where T_b is the bit period. C_{p1} , C_{p2} , L_1 and L_2 can generally be made small enough to ensure that this criteria is met for the data rates desired (as high as 11 Gb/s [84]).

The accuracy of the simplified circuit is assessed by comparing its frequency response against that of the unsimplified circuit in Figure 4.17, which shows good correspondence between the two over the frequency band of interest (i.e. below the peak in the response, which causes ISI), with error $< 10\%$ up to about 6.9 GHz.

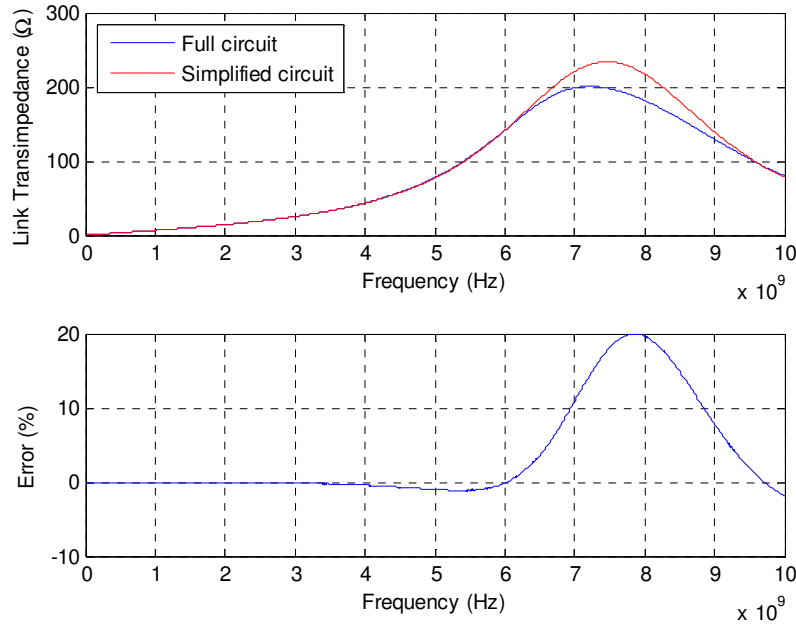


Figure 4.17: Magnitude response of circuit with full parasitic model, compared with simplified circuit using unilateralized coupled inductor model, with $L_1 = L_2 = 5$ nH, $k = 0.2$, $R_{p1} = R_{p2} = 100$ Ω , $R_{tx} = 10$ k Ω , $C_{p1} = 75$ fF and $C_{p2} = 100$ fF

4.2.1 Transmitter and receiver design

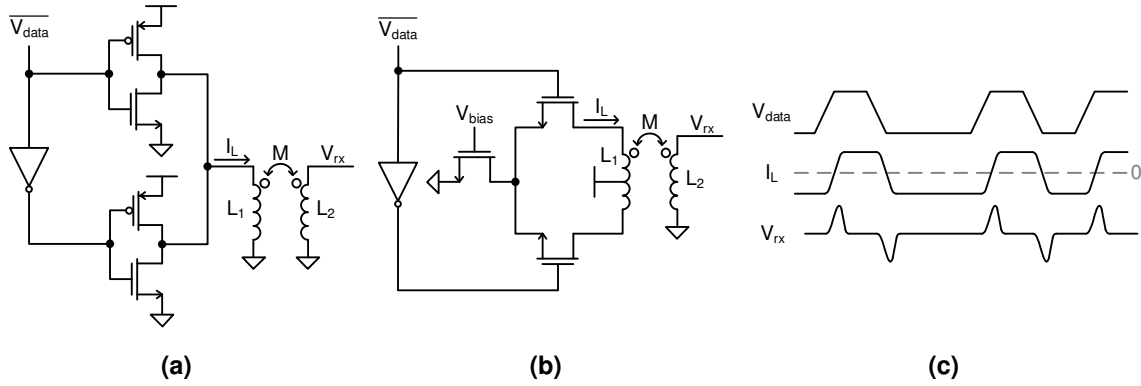


Figure 4.18: Constant-current transmitters: (a) H-bridge, (b) CML-based and (c) operation waveforms

As outlined above, careful design of the coupled inductors is necessary to ensure a clean channel response and prevent ISI. The next challenge is designing transmitter and receiver circuitry; since

the channel acts like a differentiator, a major concern is amplifying and synchronizing the resulting voltage pulses at the receiver.

The simpler form of transmitter is the constant current type. As their name implies, these designs send a current through the transmitter coil whose direction depends on the value of the bit to be transmitted; when the bit changes, the current direction is reversed and the corresponding gradient produces a voltage pulse at the receiver coil. There are two basic implementations of such a design, one of which uses complementary transistor pairs in an H-bridge configuration to switch currents (Figure 4.18(a)) [84]. The other uses a CML-style arrangement, with a tail current source and switches to direct the current flow in a center-tapped transmit coil connected to the supply (Figure 4.18(b)) [8]. In general, the complementary-pair H-bridge arrangement is preferred for its more straightforward design, but the CML-style approach finds application when NMOS devices are significantly faster than PMOS devices (or PMOS devices are simply unavailable), such as in DRAM processes [8]. The use of a tail transistor to define the output current does provide a measure of flexibility in the design, since adjusting its bias point can change the output current amplitude – useful for saving power and/or adjusting for process, voltage or temperature (PVT) variation.

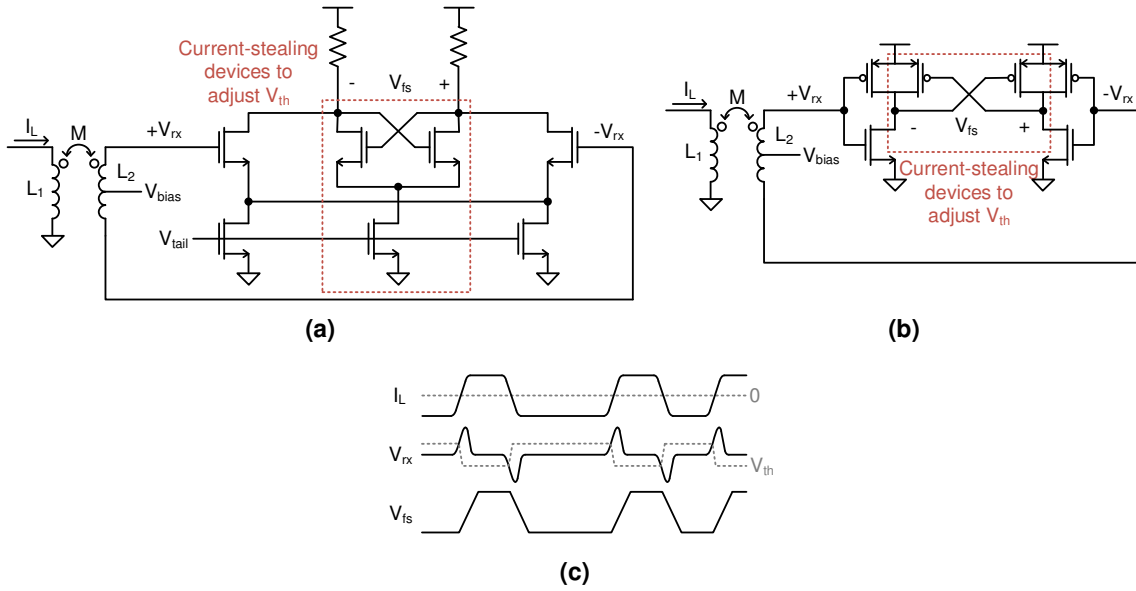


Figure 4.19: Variable-threshold comparators: (a) CML-based, (b) inverter-based and (c) operation waveforms

Since the constant-current transmitter only generates one change in current value per bit transition, the receiver likewise sees only a single pulse per bit transition. This eases timing margins by allowing the use of an asynchronous variable-threshold comparator prior to the input slicer/retiming circuitry (Figure 4.19), in a fashion similar to that used by the capacitive link in Figure 4.3. However, the constant current draw increases the power consumption of these transmitters substantially.

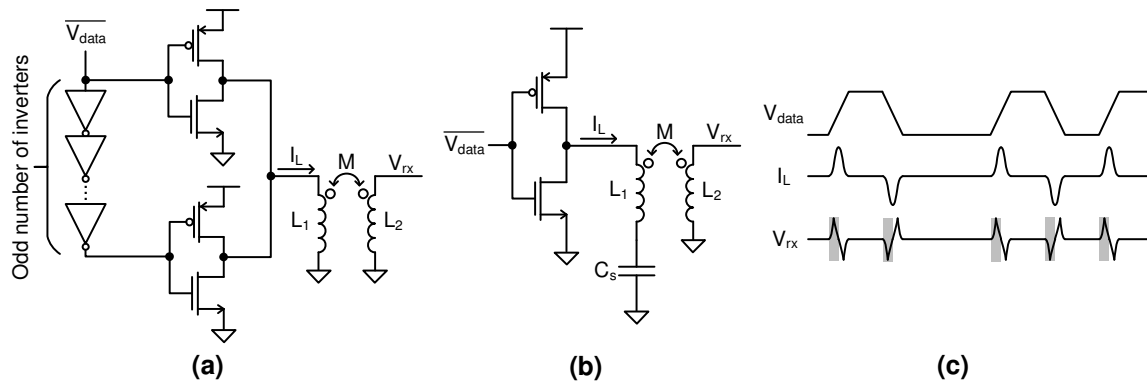


Figure 4.20: Pulsed-current transmitters: (a) H-bridge with delay line, (b) single-ended with storage capacitor and (c) operation waveforms, showing small timing margin available at receiver

The alternative is to pulse the transmitter current output, either by introducing a delay line to the H-bridge complementary pair (Figure 4.20(a)) [82], or by using a single inverter and storing the charge drawn through the inductor on a capacitor (Figure 4.20(b)) [7]. This capacitor essentially emulates the function of the second inverter in the H-bridge and reduces power consumption, but leakage limits the number of consecutive identical digits (CID, i.e. long runs of 1's or 0's) that can be tolerated.

Although the pulsed-current design does reduce the power required at the transmitter, it creates a timing issue at the receiver. Every bit transition now creates a current *pulse* instead of a current *transition*, the result of which is a pair of complementary voltage pulses at the receiver (Figure 4.20(c)). The complementary nature of the pulses (one positive, one negative) precludes the use of a variable-threshold amplifier. Instead, designs have used strongARM latches to amplify the input signal and slice it; timing the clock of these slicers so that they accurately capture the first voltage pulse is critical and difficult, given the short duration of the pulse.

The relative strengths and weaknesses of the two transmitter design approaches mean that they are each suited for different regions of operation. At lower data rates (1.25 Gb/s in [82] and 1 Gb/s in [7]), the pulsed-current approach is preferred because it offers a greater potential for power savings (pulse widths can be made a much smaller fraction of a UI than at higher data rates), and timing margins are less tight. In contrast, as data rates increase, generating and distributing clocks that can track the strict timing requirements of a pulsed-current transmitter becomes a prohibitively difficult and power-hungry exercise – to such a point, indeed, that the excess power required for clock and timing simply overwhelms any potential power efficiency that might be realized with a pulsed-current transmitter. As a result, high-data-rate inductively-coupled links have tended to use the constant-current methodology [84], [8].

4.3 Proximity Interconnect for Low-power Systems

The approaches to proximity interconnect described above have (with one exception [80]) focused exclusively on 3D integration for multi-Gb/s links in high-performance computing systems. These designs have therefore focused on achieving the highest possible data rate density (100's of Gb/s/mm²) with reasonable power consumption (as low as 1 pJ/bit [8]), and have assumed that reasonably good alignment (within 10's of μm in the case of inductive links, $< 10 \mu\text{m}$ for capacitive links) could be achieved in straightforward manner during system assembly using, for example, the technique described in [74].

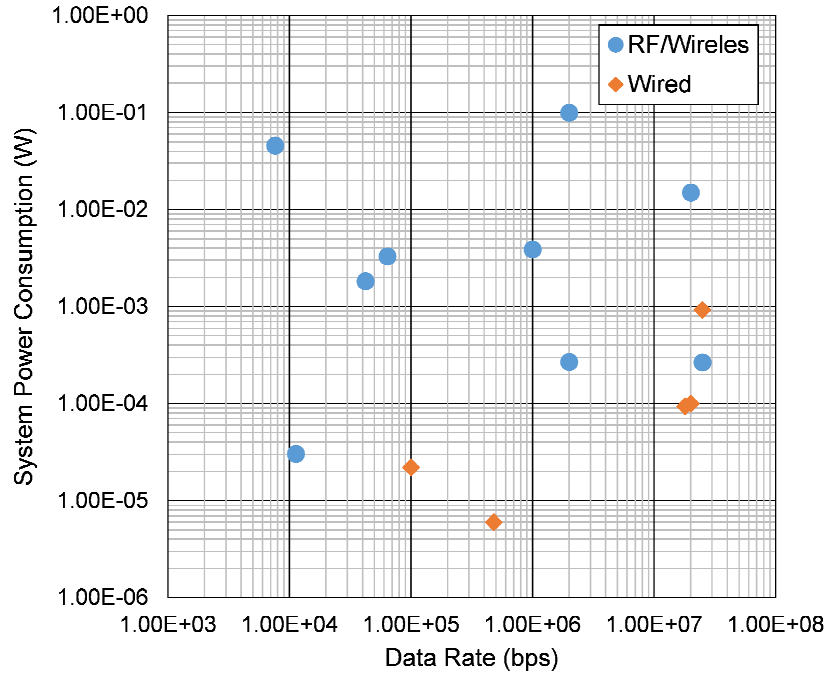


Figure 4.21: Survey of wearable and implantable biomedical devices reported in ISSCC between 2010 and 2013

Considerations for ultra-low-power, low-data-rate links such as those used in biomedical implants and ubiquitous-computing applications are quite different. A survey of implantable and wearable devices for biomedical applications reported in the International Solid-State Circuits Conference from 2010-2013 (Figure 4.21) is quite informative. The survey covers a broad range of devices (signal acquisition frontends [85], [86], [87], [88], [89], a sensing system for a tongue-computer interface [90], retinal prosthesis SoCs [91], [92], a wirelessly-powered locomotive implant [93], transcutaneous links [94], [95], an intraocular pressure sensor [96], and body-area networks for sleep monitoring [97] and ECG [98]), and shows that reasonable biomedical data loads lie in the region of 10 kb/s to 30 Mb/s, with total system power consumption < 100 mW. This stands in stark contrast with the high-performance systems mentioned previously, which typically have data rates in the 10's to 100's of Gb/s and power consumption > 100 W (Table 1.1). In this environment, energy efficiency becomes the overriding concern.

A further concern for low-power devices is mechanical alignment. Since they are intended to be cheaply assembled, many of the processing techniques that ensure good alignment in high-

performance proximity interconnect are simply uneconomical. In addition, the environments where these low-power devices find themselves can be more hostile to alignment than the typical workstation or server – vibration, tissue growth and other movement can jostle and warp the device so that alignment between chips changes considerably over time. In this context, the ability to sense changes in alignment and adapt to them becomes an even more important consideration than in the high-performance space, where the primary concern is thermal expansion (e.g. [81], where misalignment is compensated for only up to $\pm 25 \mu\text{m}$).

The twin demands for energy efficiency and misalignment tolerance can be contradictory; for instance, the need for misalignment tolerance suggests the use of inductively-coupled proximity links, since they have longer range. At <100 Mbps data rates, current-pulse-based designs are a necessity in order to save power – yet these designs tighten timing requirements considerably if the pulses are short enough to save a meaningful amount of power. As a result, complex timing recovery is required, which, compared to the straightforward timing recovery typically required for such low data rates, can negate the power saved by using pulsed-current transmitters.

To compound matters, inductively-coupled links do not lend themselves well to use as alignment sensors. The size of inductors required (e.g. $60 \mu\text{m}$ and $79 \mu\text{m}$ diameters at $110 \mu\text{m}$ pitch in [8]) is typically much larger than that for capacitive plates (e.g. $30 \times 30 \mu\text{m}$ at $36 \mu\text{m}$ pitch in [10]), and inductive coupling has better range; both these properties conspire to reduce achievable alignment-sensor resolution. The intrinsic high-pass transimpedance of an inductive link is also a drawback – the short voltage pulses produced at the receiver are inherently more difficult to quantize than the square waves produced by the relatively benign transfer function of a properly-designed capacitive link, thus complicating measurements of mutual inductance.

Capacitively-coupled proximity interconnect, on the other hand, poses much less of a power and timing problem and is much more amenable to alignment-sensing operations. However, existing approaches face a number of challenges. Specialized alignment sensors using a separate set of plates [78], [79] require more area of the two chips to be aligned (both the alignment sensor and the communication array need to be in alignment), limiting flexibility. They also require link quality to be inferred from alignment data, instead of directly measuring it, a relatively costly operation within the context of low-power devices having limited memory and processor

resources. Alignment sensors which share plates with the communication array avoid the need for link quality inference calculation and limit the area of alignment between the two chips to the communication array only. However, embedding all the alignment sensing and digitization circuitry under each plate makes plate pitch unacceptably large.

Adequately addressing the power, alignment sensing and adaptation demands of proximity interconnect for low-power devices, such as those that might be used in Brick-and-Mortar design or origami biomedical implants, requires an innovative method for embedding an alignment sensor into the communications array. Echoing the themes mentioned for the design of wireline links in the context of high-performance systems in section 2.4, multi-functional hardware that can be used for both communication and alignment sensing, and that is shared between plates wherever possible, is vital to reduce power consumption, improve robustness to misalignment and increase plate density, thus improving performance in both communication and alignment sensing contexts.

Chapter 5: Capacitive Proximity Communication for Origami Implants

Designers of medical implants face three primary challenges: size, cost and power consumption. At the same time, there is a desire for an increase in the capability of these implants – both in terms of an expansion in the scale of current functionality, such as increasing the number of electrodes in neural recording or retinal prosthesis implants, as well as adding new functionality. Size and power considerations have driven the use of specialized, highly-integrated system-on-chip designs (e.g. [86], [88], [89], [92]), which can be cost-prohibitive for the low-volume applications typical in the biomedical market. Additionally, as described in Section 1.3, increasing the scale of existing functionality can make even highly-integrated designs, such as a thousand-electrode retinal prosthesis, approach the limits of implant size in small and delicate organs such as the eye.

Given the low leakage power and high voltage tolerances typically required of biomedical implants, it seems unlikely that CMOS scaling alone can provide the size reductions required to meet future demands for increased capability; even if it can, increased development complexity and mask costs will certainly exacerbate already steep development costs and long times-to-market.

In order to address concerns of size and cost, large systems can be split into multiple chips and connected using 3D integration techniques. Previous research has demonstrated the viability of the polymer Parylene-C as a biocompatible substrate to encapsulate ICs, break-out connectivity and integrate discrete components such as capacitors [16], and its flexibility and robustness permits the construction of foldable, squashable structures such as an inductive coil [17]. It therefore provides an appealing foundation for the design of implants that can be folded compactly for implantation and then deployed into operating configuration inside the body,

minimizing the invasiveness of implantation. Additionally, this engineered or origami folding technique can be used to realize mechanically useful shapes, such as conforming a retinal prosthesis to the back of the eye, which can improve electrode contact and make stimulation more effective. This concept can be further extended to address the high cost of developing custom SoC designs for each new implant; electronics can be partitioned into commonly-used functional blocks, mass-produced as ICs that are embedded into parylene library modules. Custom implants can be assembled from these modules, reducing the cost of development and time-to-market.

Proximity communication provides a compelling way to achieve chip-to-chip communication in the context of modular origami implants. The ICs in an origami implant can be placed face-to-face, across folds and between modules, and the plates or inductors forming the coupled link can be realized using the existing metal stack, so the cost of integration is low. Since the origami implant should be as simple as possible to assemble and is deployed in-body, the alignment between the communicating chips will be poorly controlled, and will change over time due to patient movement and tissue growth. The requirements for extremely low power, alignment sensing and adaptation to alignment discussed in Section 4.3 are relevant to the design of proximity interconnect for origami implants, and the lower power, greater density and amenability of capacitively-coupled proximity interconnect to communication-array-embedded alignment-sensing make it the preferred choice for this application.

5.1 Plate and array design

The proximity interconnect is formed by capacitive coupling between plates in the pad-level (topmost) metal layer of the two chips. In order to maximize flexibility, both sides of the link implement full receiver and transmitter functionality. However, since only one side of the link needs to be able to sense alignment, two different array types, sensor and target, are used. Making a distinction between array types confers important advantages: it allows the sensor array to be separately optimized from the target array to increase alignment-sensing resolution,

and saves power and area by eliminating unneeded sensing circuitry from the target array. No significant design time is added, since the target array uses a subset of the sensor array blocks.

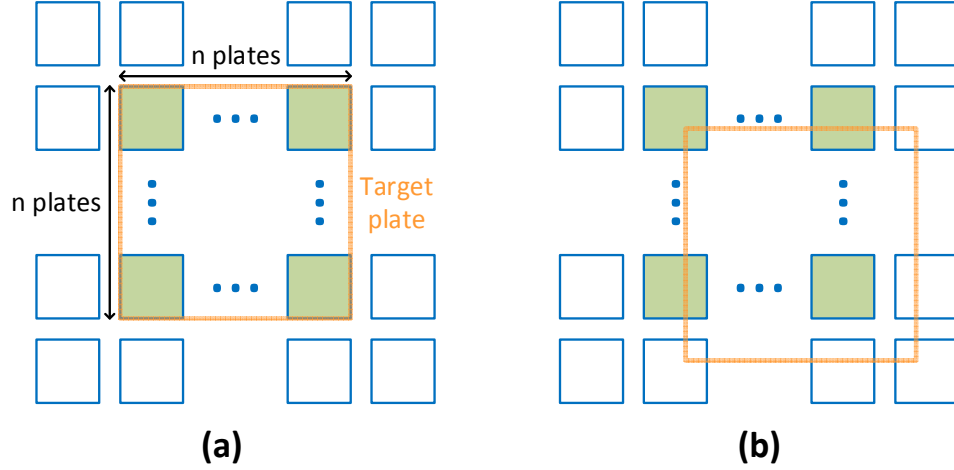


Figure 5.1: Top-down view of the sensor array, in (a) best-case (maximum overlap) and (b) worst-case (minimum overlap) alignment. Active sensor plates are shaded in green, and the associated target plate is outlined

The sensor array is split up into smaller constituent plates, with n^2 plates corresponding in size to one target plate (Figure 5.1). These smaller plates are joined together in groups, which form the basic units of alignment sensing and communication and are analogous to pixels in an image sensor. This compound-plate configuration enhances alignment-sensing resolution by reducing the effective sensing-unit (‘pixel’) step-size to the dimensions of the smaller plates. The smaller step-size is also beneficial during communication, since it increases the overlap area between sensor and target in the event the two are misaligned. There are limits, however, to how small the constituent plates can be made. As these plates are made smaller, more of them need to be connected together to equal the size of the target plate, increasing switch and routing capacitance and reducing link gain. Additionally, the gap in-between each sensor plate is governed by the minimum pad-level metal spacing allowed in the design rules; smaller constituent plates result in more area lost to these gaps, reducing parallel-plate coupling capacitance, an effect only partially mitigated by the corresponding increase in fringing capacitance.

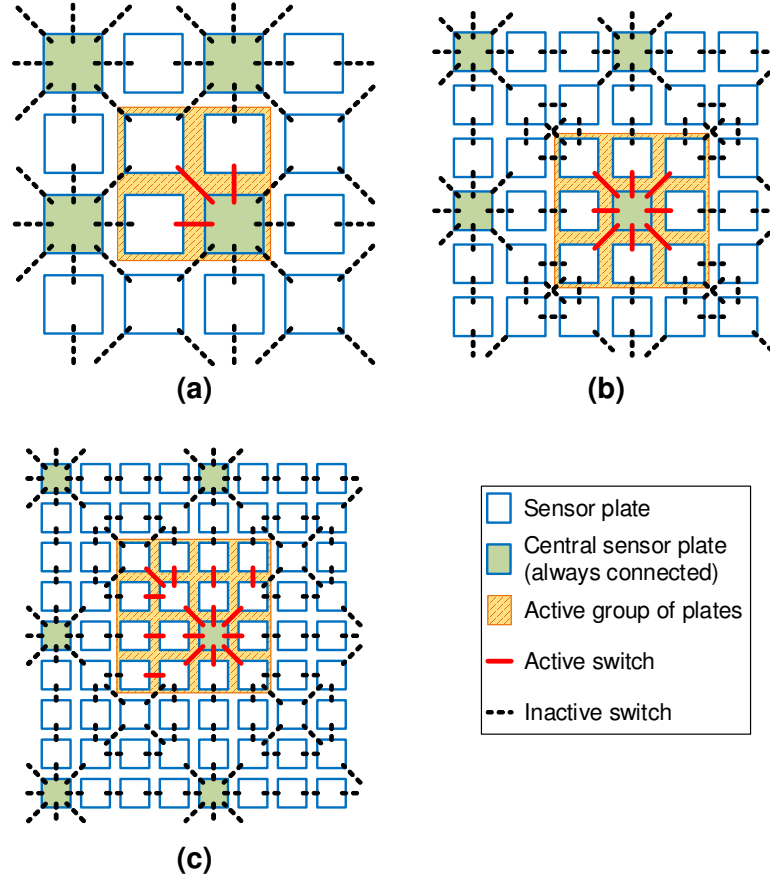


Figure 5.2: Plate and switch configurations for (a) $n = 2$, (b) $n = 3$ and (c) $n = 4$. The highlighted group of active sensor plates indicates the worst-case loading condition, where the largest number of inactive switches are connected to the active plates

Assuming that biasing resistance (R_{bias}) and transmitter drive resistance (R_{tx}) are appropriately sized and the ‘on’ resistance of the switches used to connect the plates is small, link gain can be expressed by a capacitive voltage divider, modified from (4.1):

$$\frac{V_{rx}}{V_{tx}} = \frac{C_C}{C_C + C_G + C_{sw} + C_w + C_{rx}} \quad (4.19)$$

where C_G is the intrinsic parasitic capacitance of the sensor plates to the ground plane, C_{sw} is the switch parasitic capacitance, C_w is the wiring parasitic capacitance and C_{rx} is the input parasitic capacitance of the receiver. The receiver design is not directly affected by the choice of n , so C_{rx} can be assumed fixed with respect of n . Switch capacitance can be estimated by:

$$C_{sw} \approx C_{on} \cdot sw_{on} + C_{off} \cdot sw_{off} \quad (4.20)$$

where C_{on} is the parasitic capacitance of an ‘on’ switch, and C_{off} is that of an ‘off’ switch. Using the plate and switch configurations in Figure 5.2, the number of active switches necessary to connect n^2 plates to the central, always-connected plate(s) can be expressed by:

$$sw_{on} = n^2 - 1 \quad (5.1)$$

That is, one for every active plate except the central one, which is always connected. The number of inactive switches loading the group of plates (sw_{off}) is somewhat more difficult to determine, since it depends on the exact location of this group within the array. Even considering only the worst-case configuration, it is difficult to derive a closed-form expression for sw_{off} across all values of n . However, because the routing complexity and number of switches added for $n > 4$ rapidly approaches the impractical, it is sufficient to express sw_{off} as a look-up table for $n \leq 4$ (Table 5.1).

n	sw_{off} (worst-case)
1	0
2	10
3	16
4	18

Table 5.1: Number of inactive switches loading the active group of plates, for various values of n

Wiring parasitic capacitance can be estimated as:

$$C_w \approx (sw_{on} + sw_{off}) \cdot \frac{C_{w,unit}}{n} \quad (5.2)$$

where $C_{w,unit}$ is the capacitance of a wire running the whole length of the plate group. For simplicity, all plate-to-plate wires are assumed to present the same parasitic load.

The optimal value of n can be selected by running different configurations through a 3D EM field solver to extract C_c and C_g . Results for the dielectric and metal configuration in Figure 5.5, with a 12 μm parylene dielectric interposed, $C_{on} = 2 \cdot C_{off} = 1$ fF, $C_{w,unit} = 10$ fF and $C_{rx} = 5$ fF are shown in Figure 5.3 and Figure 5.4. Under best-case alignment, increasing n does not cause an increase in overlap between the target plate and the groups of sensor plates, so the extra parasitics and reduction coupling capacitance results in a direct loss of gain. However, under

worst-case alignment conditions, better gain is realized when $n = 2$, due to the increase in overlap between target and sensor. However, no further benefit in either alignment condition is realised by increasing n to 3; in fact, the extra parasitic loading results in a *decrease* in link gain under best-case alignment. In order provide reasonable margins for noise and input slicer offset, a link gain >0.03 (corresponding to a received signal amplitude >30 mV_{PP} for an input amplitude of 1 V_{PP}) across a 12 μm parylene dielectric (the thickest tested) was targeted. Therefore, $n = 2$ and a target plate dimension of 60x60 μm were chosen for this design (Figure 5.5).

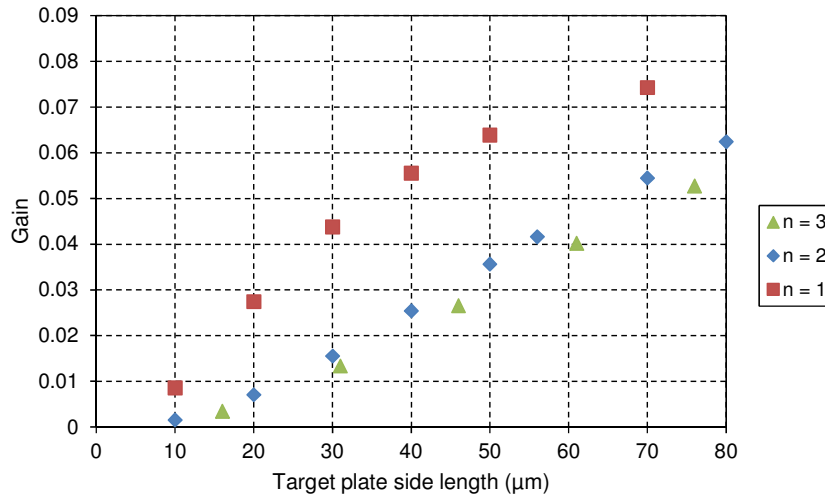


Figure 5.3: Link gain for different values of n , when plates are in best-case alignment

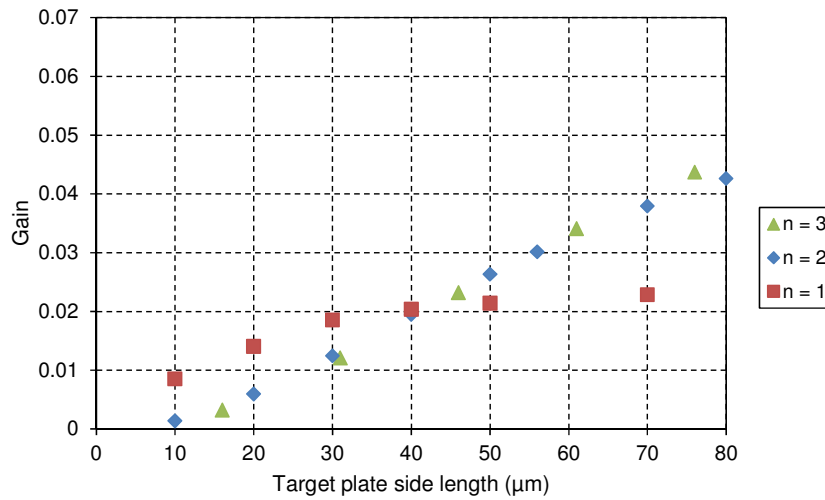


Figure 5.4: Link gain for different values of n , when plates are in worst-case alignment

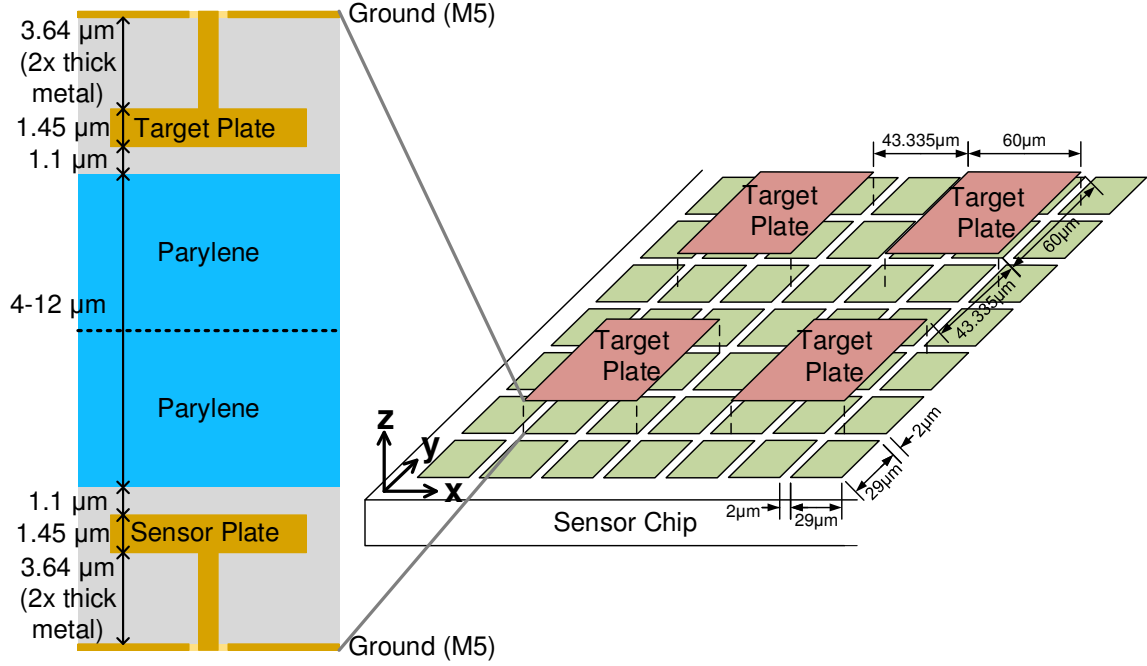


Figure 5.5: Dielectric and metal layers used to form plate structure, and sensor/target array arrangement (target chip outline not shown for clarity)

In order to ensure a successful connection is made between the two chips, a certain amount of overprovisioning is necessary; making the target and sensor arrays larger than the minimum necessary increases the likelihood that a sufficient number of plates are coupled strongly enough to support the desired data rate, even in the presence of poor alignment. The array adaptation logic can select as many of the best-coupled, most efficient sets of plates as necessary to support the target data rate, turning off the remaining, less-efficient (or even completely uncoupled) plates to save power. To provide further flexibility, the communication link is made bidirectional, with either the target or sensor plates capable of operating as the transmitter.

While selecting $n = 2$ does improve link gain in the case where the target and sensor plates are poorly aligned, a decrease in gain still results (e.g. from about 0.047 to about 0.034 when the target plate size is $60 \times 60 \mu\text{m}$), which affects the noise margins, power efficiency and maximum achievable data rate across the link. If, due to in-plane (x- and/or y-axis) misalignment, all the target plates are simultaneously poorly aligned to the sensor array, the array adaptation scheme will have no choice but to select poorly-coupled, less efficient links (and more of them) to make up the required data rate. This situation can be avoided if the target plates are spaced at a non-

integer multiple of the sensor plate pitch. Even if the target plates can therefore never all be simultaneously perfectly aligned with the sensor, neither can they all be in worst-case alignment, providing some flexibility for the adaptation scheme to pick the best-coupled sets of plates. For this design, a multiple of $3\frac{1}{2}$ is chosen (Figure 5.5).

An additional concern is crosstalk between adjacent target plates. Although crosstalk in capacitive proximity interconnect is less of a concern than in inductive proximity interconnect, it can become significant if the target plates are close enough to each other. High communications density can be maintained, even in the face of significant potential crosstalk, by using differential signalling and careful arrangement of the transmitting plates [10]. However, differential signalling has the disadvantage of requiring two, instead of one, set of plates to be well-coupled, complicating the alignment problem. The additional energy required for differential signalling is also an issue given the power constraints imposed in biomedical implants. Instead, the spacing in-between target plates is made relatively large in order to prevent crosstalk, which a field-solver simulation suggests will be negligible for the configuration used.

5.2 Transceiver array with distributed alignment sensing

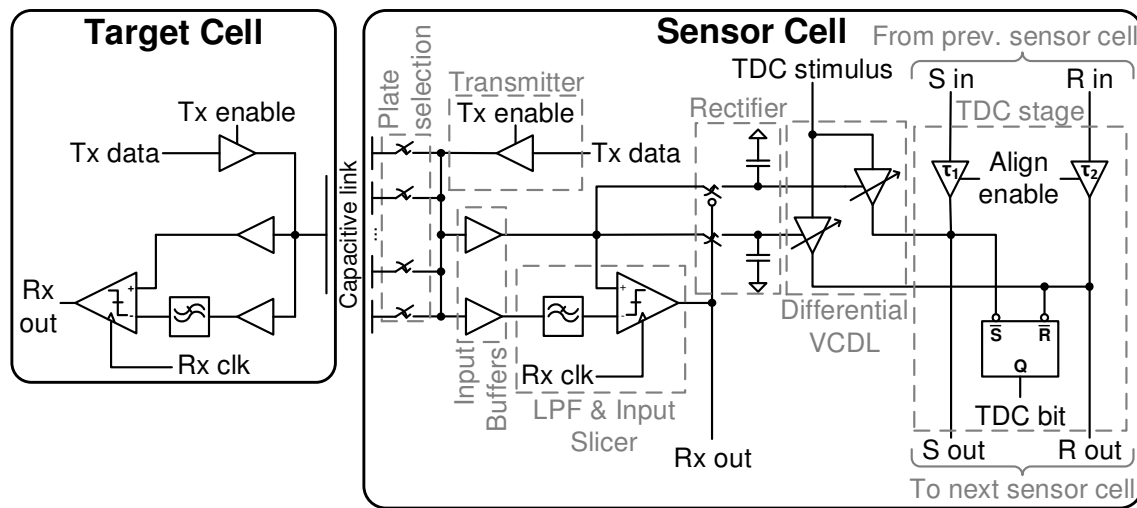


Figure 5.6: Architecture of the sensor and target cells, with key functional blocks indicated

The structure of the target array is relatively straightforward – each target plate is uniquely associated with a single target cell, which contains the transmitter and receiver for that plate. On the other hand, because the sensor array adds the ability to conduct alignment sensing and each sensor cell can associate itself with one of 4 possible groups of 4 sensor plates, sharing these plates with its neighbours (control logic ensures that no plate is connected to more than one cell at a time), the structure of the sensor cell is somewhat more involved.

The architecture of individual target and sensor cells is shown in Figure 5.6. The target cell is a subset of the sensor cell, and the two share the same transmitter and receiver (composed of a low-pass filter and input slicer) designs. For alignment sensing, the sensor cell adds a rectifier, differential VCDL and time-to-digital converter (TDC) stage; these components and the alignment sensing methodology are discussed in more detail in sub-section 5.2.1, below.

5.2.1 Alignment sensing

The dielectric between the two plates is composed of the passivation as well as one or two parylene sheets, each 4-6 μm thick, depending on the exact structure of the parylene module. Since the distance between the two plates can be relatively large compared to the distance between each plate and its corresponding ground plane (Figure 5.5), it is difficult to distinguish the capacitance between the sensor and the target plate from that between the sensor and the ground plane *under* the target plate. The ground plane could be moved to a lower level metal under the target plate, but the restriction this would place on routing density ultimately makes this option unacceptable. Although this prevents the use of techniques that attempt to sense an undriven capacitance in order to determine alignment [80], the restrictions that such techniques attempt to address (an unpowered target chip) do not apply in this case. Instead, it makes sense to use the existing transmitter circuitry in the target cell to drive a stimulus (e.g. an alternating sequence) onto the target plate to electronically distinguish it from the ground plane. The amplitude of the signal received at the sensor is proportional to the amount of coupling between the plates, and can be used to determine the link quality and alignment of the two chips.

In order for this coupled amplitude to provide useful information for an array adaptation scheme, it will have to be digitized. Placing a full ADC under each group of sensor plates is

unappealing from a power and area standpoint, especially if all the sensor cell circuitry (switches, transmitter, receiver, alignment sensor, digital control) is to fit under a single set of sensor plates (a roughly $60 \times 60 \mu\text{m}$ area). Instead, a distributed approach needs to be taken, where elements or stages of the ADC are split up and spread throughout the sensor array. Although this restricts the number of groups of sensor plates that can be sensed simultaneously, determining the alignment is not a time-sensitive operation and the trade-off to save power and area is an acceptable one.

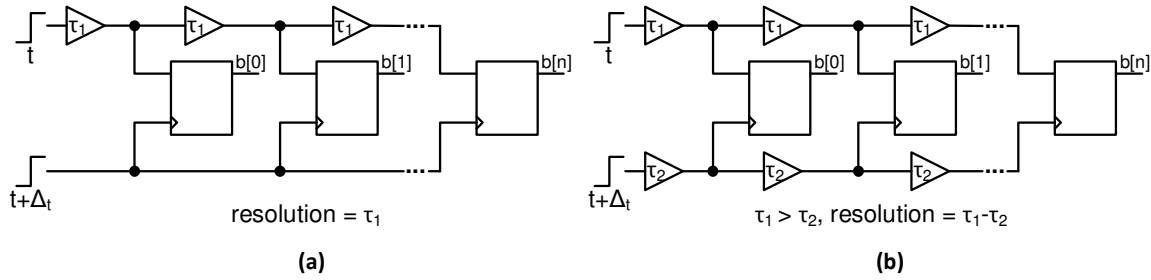


Figure 5.7: (a) Standard TDC using a single delay line compared to (b) a vernier TDC. Flip-flops act as arbiters, indicating which edge arrives earlier

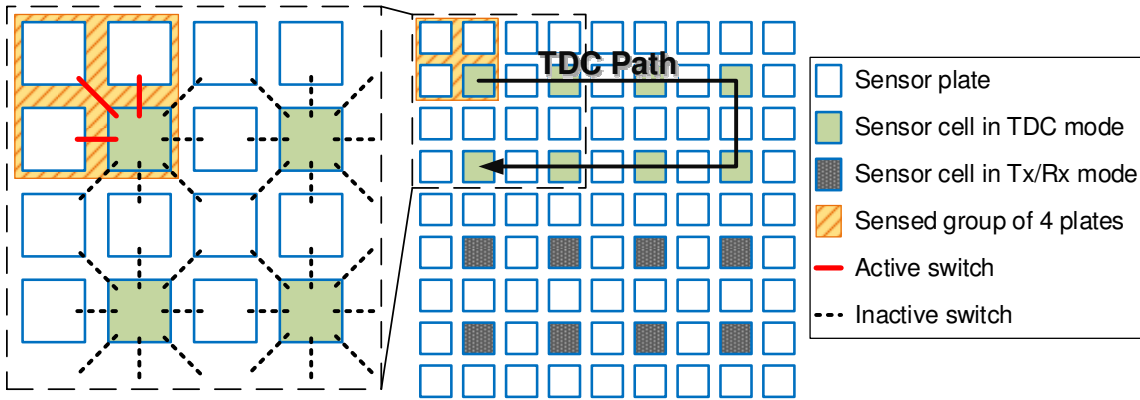


Figure 5.8: Sensor array structure, showing TDC path for alignment sensing at indicated plates

Thanks to its inherently segmented nature, good resolution characteristics and low power consumption, a TDC is an appealing candidate for use as the digitizing element in the alignment sensor. The most straightforward version of a TDC uses a delay line with a set of flip-flops connected at the output of each delay element (Figure 5.7 (a)). The delay elements are typically realized using inverters, so the resolution of such a delay line is, in theory, limited to the shortest delay achievable by a single inverter stage. However, this limit is difficult to achieve in practice

due to the presence of wiring and device parasitics; additionally, since the converter resolution is dependent on the absolute delays of a set of inverters, PVT variation can have a significant effect on performance. Most of these problems can be avoided by defining the resolution as the difference between delay elements, in a vernier TDC (Figure 5.7 (b)) [99]. Each stage of the TDC (set of delay elements and arbiter) can be distributed through the sensor array and connected together during alignment sensing to form a complete TDC; in this design, a 7-stage TDC is used for 3-bit output (Figure 5.8).

There still remains the problem of converting the amplitude of the signal coupled from the target to the sensor into a delay, which the TDC can use to generate a digital word. The amplitude can be converted into a DC level using a rectifier, and this voltage used to bias a VCDL to generate the delay. The receiver contains an input slicer, which makes decisions about the bit value of the received signal; this same slicer can be used to drive a rectifier (Figure 5.9). When the output of the input slicer transitions, the rectifier generates pulses to control switches that shunt the high or low levels of the received signal ('in') onto the appropriate storage capacitor. To ensure that the capacitors capture the correct values, the delay from a transition in the received signal to the pulse, and the pulse itself, must be short enough so that the pulse ends before a further transition in the received signal:

$$t_{\text{slicer}} + t_{\text{pg}} + t_{\text{pw}} < t_{\text{bit}} \quad (5.3)$$

To increase timing margin, the target cell transmitter is set to output a low-frequency (quarter-rate) alternating sequence during alignment sensing.

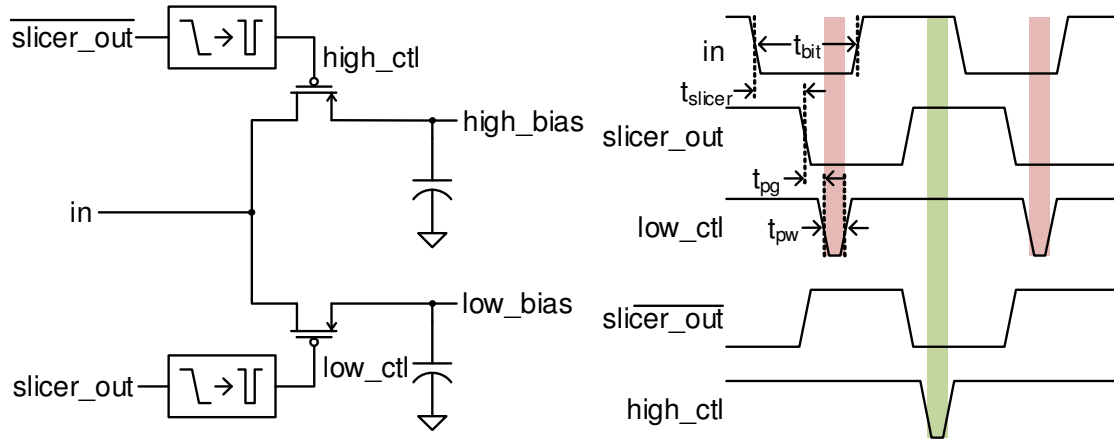


Figure 5.9: Rectifier and associated timing diagram

The rectified voltages are used to bias a differential VCDL (Figure 5.10). The bias voltage adjusts the pull-down strength of an inverter in each unit cell; higher voltages result in stronger pull-down and, because the TDC stimulus is a rising edge, less delay. The link quality is assessed by comparing the TDC output code against a look-up table of supported data rates; similarly, alignment is determined by comparing the TDC output against results from the field solver simulation. As a result, the linearity of the ADC formed by the VCDL and TDC (whether INL or DNL) is a secondary concern compared to its total error, the deviation of the ADC's actual conversion characteristic away from that expected. Simulations of the implemented design suggest that offset is the most significant effect contributing to the total error, and this is corrected via variable-threshold buffers at the output of the VCDL. The threshold of these buffers is varied by digitally adjusting their pull-up strength, in thermometer-coded fashion to ensure monotonicity. Higher pull-up strength results in a higher threshold voltage, delaying the corresponding VCDL output edge to compensate for offset in either the VCDL itself or the following TDC. The amount of correction is determined by observing the zeroth TDC bit, which is the output of an arbiter connected directly to the VCDL, prior to any TDC delay elements.

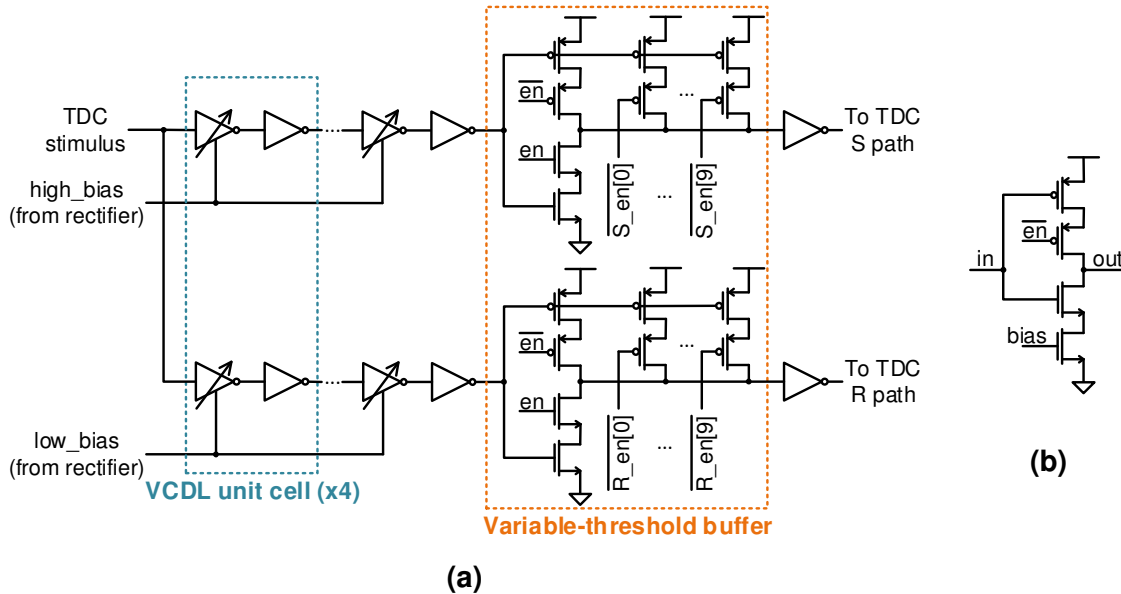


Figure 5.10: (a) Differential voltage-controlled delay line with variable-threshold output buffer and (b) variable-delay inverter used in VCDL unit cell

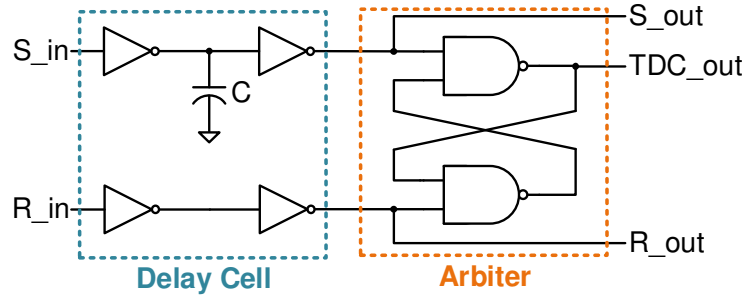


Figure 5.11: TDC delay cell and arbiter

The TDC arbiter is an SR-latch, selected for the symmetric loading characteristics it presents on its inputs. The delay element is constructed from a matched set of inverters on both the S and R paths, with a metal capacitor added to the S path so that $\tau_1 > \tau_2$ (Figure 5.11). Assuming that the inverters in the two signal paths can be made to match reasonably well, the linearity of the TDC is controlled primarily by the value of this capacitor. For this reason, the lower variability of a metal capacitor is appealing; its lower density (compared to a diffusion or MOS capacitor) is not a major concern, since only a small (several fF) capacitance is required.

Vertical (z-axis) separation between the sensor and target arrays can be measured directly from the corresponding sensor cell's TDC output. The amount of coupling capacitance is inversely proportional to the distance between the sensor and target plates - when they are close to each other, coupling is high and the amplitude of the coupled signal is large, conversely, plates far apart will be poorly coupled and the amplitude will be small. The inverse relationship means that the conversion from physical separation to coupling amplitude (therefore, TDC output word) is non-linear. As a result, alignment sensor resolution is better when the two chips are close to each other, and degrades as they get further apart. This non-linearity is particularly pronounced when there is no parylene dielectric between the two chips (Figure 5.12); adding parylene in-between them limits the maximum coupling capacitance achievable, and linearizes the results somewhat (Figure 5.13).

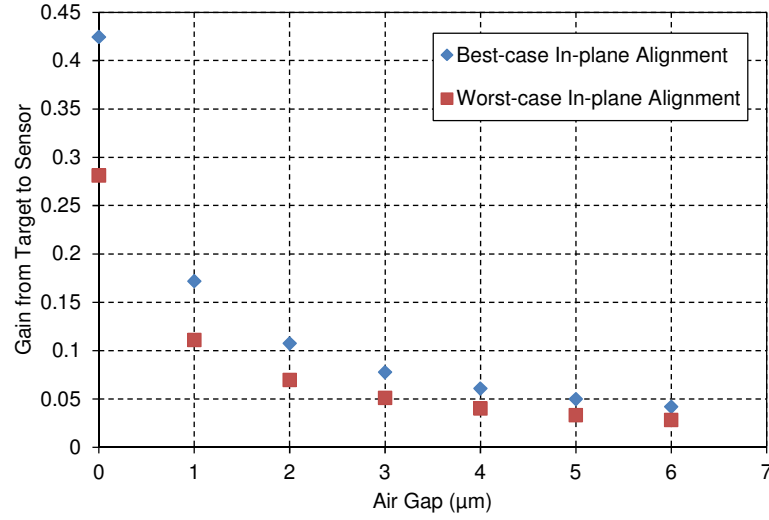


Figure 5.12: Target-to-sensor link gain vs. air gap, using $n = 2$, target plate size of $60 \times 60 \mu\text{m}$ and no parylene

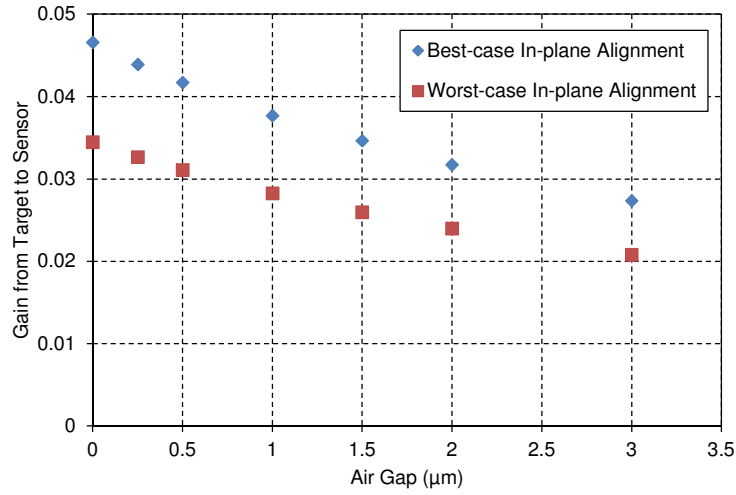


Figure 5.13: Target-to-sensor link gain vs. air gap, using $n = 2$, target plate size of $60 \times 60 \mu\text{m}$ and $12 \mu\text{m}$ parylene

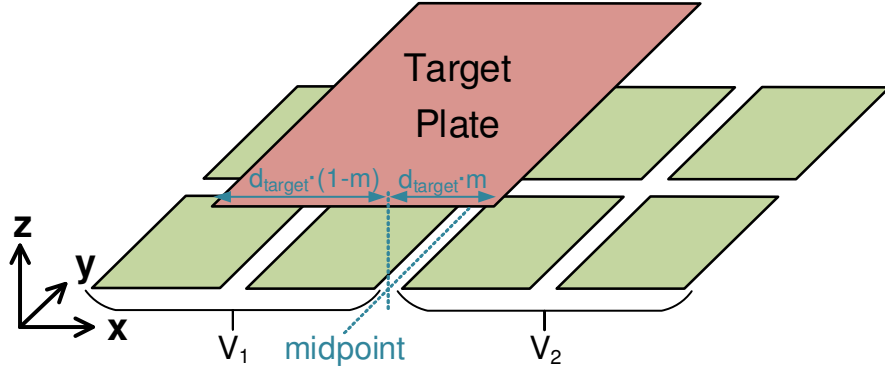


Figure 5.14: Two adjacent groups of sensor plates used for x-axis alignment sensing. $0 \leq m \leq 1$; when $m = 0$, target plate is all the way to the left (completely over V_1). Likewise, when $m = 1$, target plate is completely over V_2

Determining in-plane (x- and y-axis) alignment is somewhat more involved. The output of a single group of sensor plates is insufficient to determine the in-plane alignment of the associated target cell. Instead, readings from two adjacent groups of sensor plates are used (Figure 5.14). Ignoring fringing fields, the voltage seen at each group of plates is:

$$\frac{V_1}{V_{tx}} = \frac{\eta \cdot C_c \cdot m}{\eta \cdot C_c \cdot m + C_g}, \quad \frac{V_2}{V_{tx}} = \frac{\eta \cdot C_c (1 - m)}{\eta \cdot C_c (1 - m) + C_g} \quad (5.4)$$

where C_c is the amount of coupling capacitance between the target plate and a group of sensor plates in perfect alignment (no air gap, target plate directly on top of sensor plates), η is a de-rating factor to account for vertical separation between the plates and C_g is the parasitic ground capacitance seen at each group of plates. Taking the ratio of the two expressions yields:

$$\frac{V_1}{V_2} = \frac{\eta \cdot C_c \cdot m}{\eta \cdot C_c (1 - m)} \cdot \frac{\eta \cdot C_c (1 - m) + C_g}{\eta \cdot C_c \cdot m + C_g} \quad (5.5)$$

Assuming $C_g \gg \eta \cdot C_c$ (a reasonable assumption, since, as seen in section 5.1, link gains are typically $\ll 1$), this simplifies to:

$$\frac{V_1}{V_2} = \frac{m}{1 - m} \quad (5.6)$$

Re-arranging to find m :

$$m = \frac{V_1}{V_1 + V_2} \quad (5.7)$$

This can be converted into an offset from the midpoint between the two groups of plates:

$$d = k \cdot d_{target} \cdot m \quad (5.8)$$

where d_{target} is the side-length of the target plate and k is a scaling factor applied to correct for the effect of fringing fields, which was ignored in the derivation above. Simulating this using the implemented geometries (target and sensor plates in the top-level metal, with $d_{target} = 60 \mu\text{m}$ and $k = 1.19$) yields the results shown in Figure 5.15.

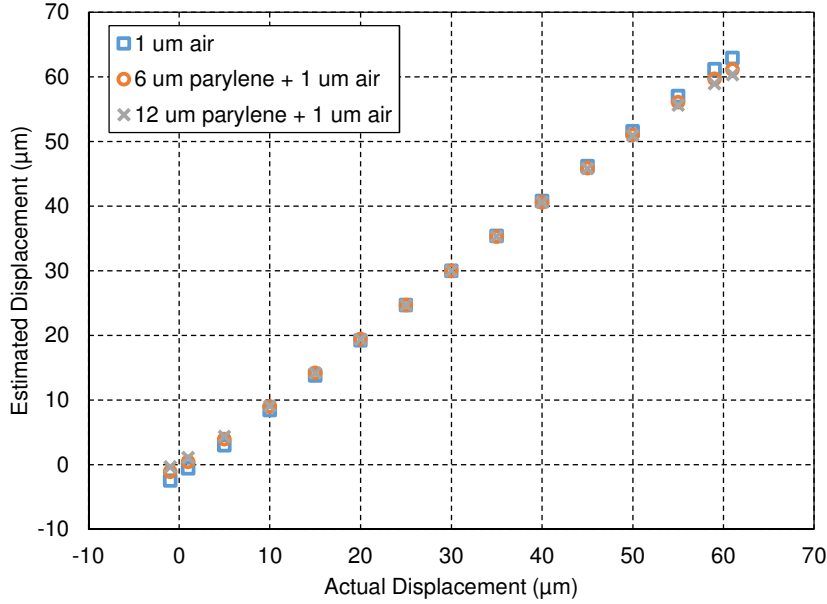


Figure 5.15: In-plane alignment estimation, using simulation results with various dielectrics between the two chips

The results shown here use the analog value of the coupled signal amplitude to calculate the displacement, and do not account for the effect of quantization due to the VCDL/TDC-based ADC. Additionally, despite the introduction of k , the effect of fringing fields is non-linear and is not completely corrected for. As a result, a small amount of non-linearity remains when the target plate displacement is near 0 and 60 μm . Note, however, that when the target plate displacement is at these extremes, the alignment sensor can simply switch to the set of sensor plates adjacent to those used to generate V_1 and V_2 . Since the target plate will be nearer the middle of the measurement range for these plates, the non-linearity is, for practical purposes, not a concern.

5.2.2 Transmitter and receiver

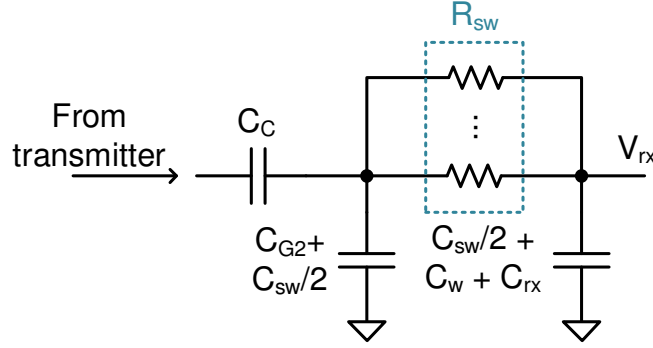


Figure 5.16: Equivalent circuit of the capacitive link, with switch parasitics (R_{sw} and C_{sw}) introduced. R_{bias} is assumed to be very large, and is omitted

In order to select the correct group of 4 plates, the sensor cell contains a bank of 8 switches (connected to the plates as indicated in Figure 5.2(a)). The resistance of these switches (Figure 5.16) adds a third pole to the link transfer function, in addition to the two already identified in (4.8):

$$\omega_{p,sw} \approx \frac{C_{G2} + C_{rx} + C_w + C_{sw}}{R_{sw} \left(C_{G2} + \frac{C_{sw}}{2} \right) \left(C_{rx} + C_w + \frac{C_{sw}}{2} \right)} \quad (5.9)$$

where R_{sw} is the equivalent resistance of all 3 active switches. Since the switches are in parallel with each other, R_{sw} is one-third the on resistance of a single switch. To account for this extra pole, the switches will simply need to be sized large enough (equivalently, R_{sw} made small enough) so that $\omega_{p,sw}$ is much higher than the Nyquist frequency of the highest data rate to be supported. For this design, with data rates < 100 Mbps and Nyquist frequency < 50 MHz, this is easily achievable; assuming $C_{G2} \approx 80$ fF, $C_{sw} \approx 16$ fF, $C_{rx} \approx 5$ fF and $C_w \approx 40$ fF, then $R_{sw} < 90$ k Ω – even minimum-size, regular threshold voltage, 65 nm PMOS devices exceed this requirement comfortably.

The input buffer, transmitter and receiver designs are used in both the sensor and target cells. The transmitter is a tri-state buffer modified with a leakage path (Figure 5.17). Low-leakage standard threshold voltage, low-power (SVTLPL) devices are used in the output path to reduce leakage power, while higher-leakage low threshold voltage, general-purpose (LVTGP) devices are used to define the source-follower input bias, in a manner similar to that described in Figure 4.5.

A leakage cut-off device is used to shut the leakage path down when the transmitter is active, to prevent unnecessary added power consumption and static current draw.

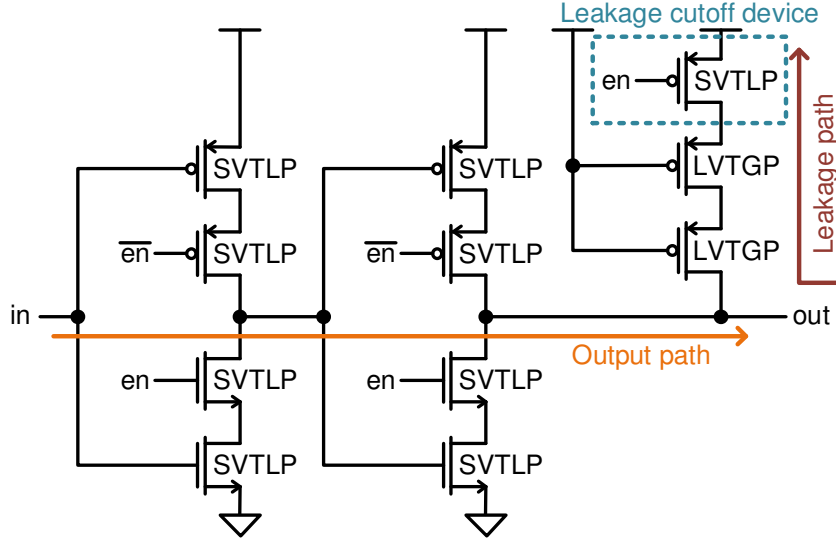


Figure 5.17: Tri-state buffer-based transmitter, with leakage path to define plate bias voltage

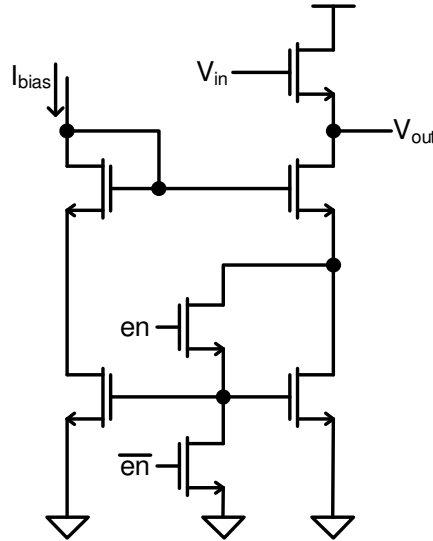


Figure 5.18: Source-follower buffer with gateable Wilson current mirror bias

The source-follower input buffers isolate the plates from the rest of the cell circuitry in order to minimize parasitic loading of the capacitive link (Figure 5.18), and drive two distinct signal paths – the first contains no filtering, and is used by both the input slicer and the rectifier. The second path contains a LPF, which generates a reference voltage for the input slicer. The bias

points of these two paths need to match well in order to ensure that the LPF generates an accurate reference voltage. Variation in the bias point is minimized through the use of long-channel devices and a Wilson current mirror to boost output resistance. The current mirror is designed to shut off when the buffer is idle (e.g. the cell is completely off or is acting as a transmitter) to save power.

In order to generate a stable reference voltage for the input slicer, the LPF has to have a cut-off frequency below the fundamental of the longest expected data sequence. For example, if the link is to be tested with a PRBS:

$$f_{lpf} < \frac{f_{data}}{l_{PRBS}} \quad (5.10)$$

where f_{data} is the data rate and l_{PRBS} is the PRBS length. Based on this equation, if the link was designed to handle a PRBS-7 at data rates as low as 20 Mbps, $f_{lpf} < 156$ kHz. Such low cut-off frequencies are difficult to achieve using purely passive elements in a reasonable amount of space on-chip. At the same time, the filter needs to accept an input signal at the data rate, so using a purely switched-capacitor approach would require a high switching frequency, wasting power. Instead, a hybrid multi-stage LPF is used (Figure 5.19). The first stage is a simple 1st-order RC filter with a relatively high cut-off frequency; the following switched-capacitor stages step down in cut-off frequency until $f_{lpf} = 130$ kHz. In order to save power and area, buffers between the various stages of the filter are omitted; the inaccuracies introduced by this omission are mitigated by the fact that the input capacitors of the two switched-capacitor stages are much smaller than the output capacitors of the preceding stage. In any case, the desired output is a simple DC bias, so any distortion resulting from the lack of stage-to-stage isolation is tolerable.

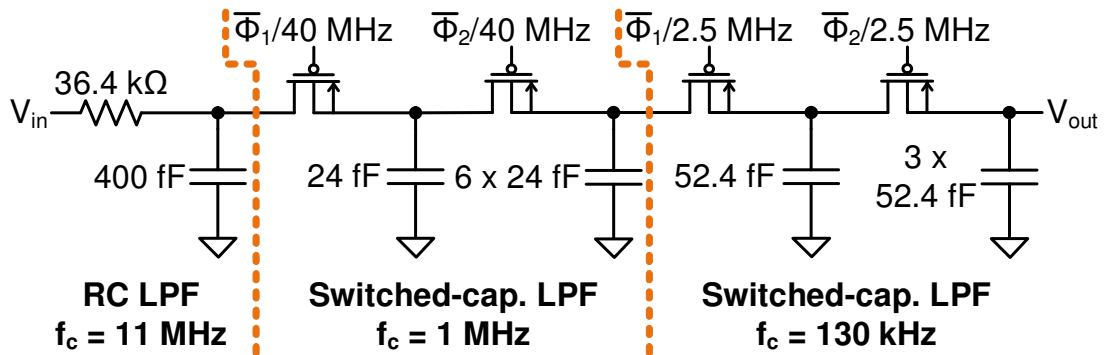


Figure 5.19: 3-stage hybrid low-pass filter

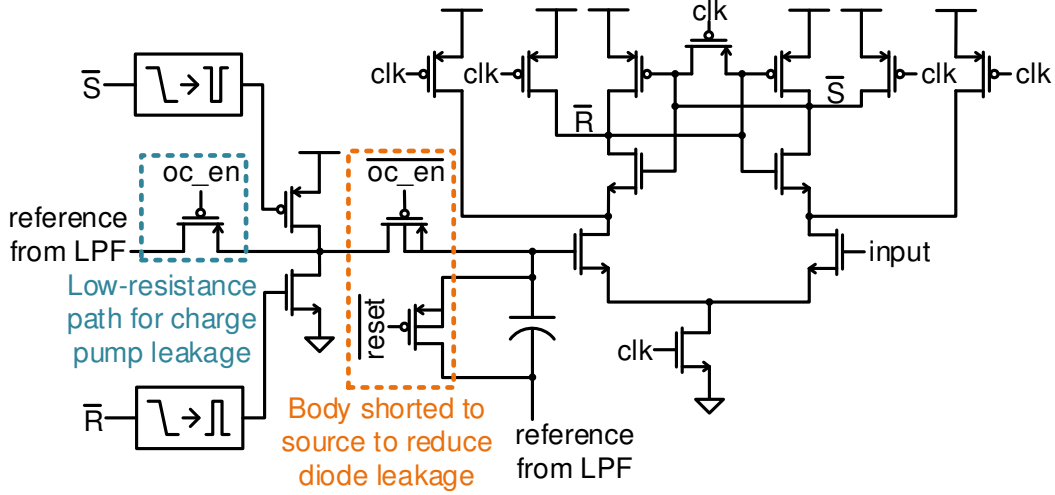


Figure 5.20: Input slicer and offset compensation (SR latch not shown). ‘Reset’ zeroes the offset compensation capacitor and ‘oc_en’ is asserted during offset compensation calibration

The input slicer is a strongARM latch used as a comparator (Figure 5.20), followed by an SR latch. Data rates < 100 Mbps are targeted, with signal amplitudes as low as 30 mV_{PP} in the event that alignment between the plates is poor. The low data rates allow the slicer plenty of time to evaluate small incoming signals, but any input offset could result in an erroneous decision. As a result, some form of offset compensation is an important part of the input slicer design. A 200-run Monte Carlo simulation of the input slicer suggests that the offset has $\sigma \approx 15.4$ mV (Figure 5.21(a)), so correcting up to 3σ offset would require the offset compensation to have a range of approximately ± 50 mV. In order to provide this range with < 5 mV resolution within a limited power and area budget, charge pump-based offset compensation is added in series with the threshold-generating LPF. Leakage is reduced through the use of a thick-oxide storage capacitor, triple-well devices to eliminate diode leakage through the switches, and the provision of a low-resistance path to shunt leakage from the charge pump switches away from the storage capacitor. These measures limit charge pump leakage to about 1 mV/ms in the FF corner. The storage capacitor does still require a periodic refresh of its stored voltage, which can be done when the link is taken down to re-acquire chip-to-chip alignment. Monte Carlo simulation of the compensated input slicer suggests that the offset compensation is successfully meeting its 5 mV residual offset target, with maximum observed post-compensation offset of about 3.5 mV over 200 runs (Figure 5.21(b)).

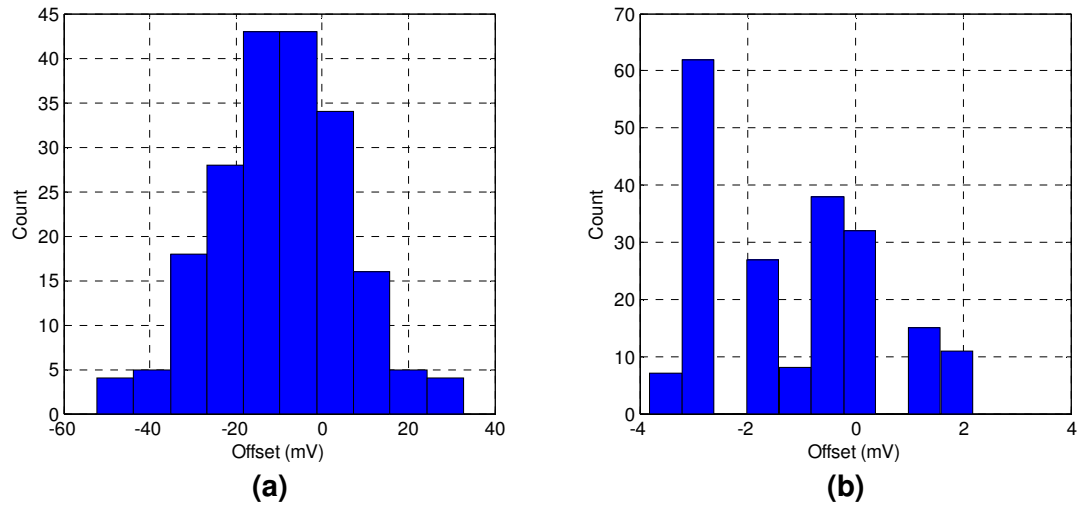


Figure 5.21: Input slicer offset estimated across 200 Monte Carlo simulation runs, (a) before and (b) after offset compensation

5.3 Hardware measurements

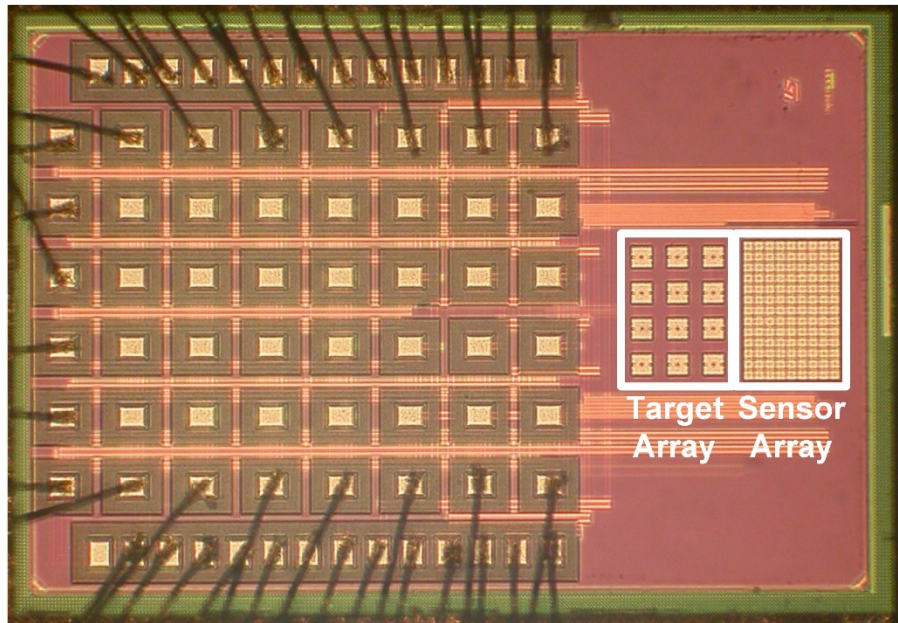


Figure 5.22: Die micrograph, with sensor and target arrays marked

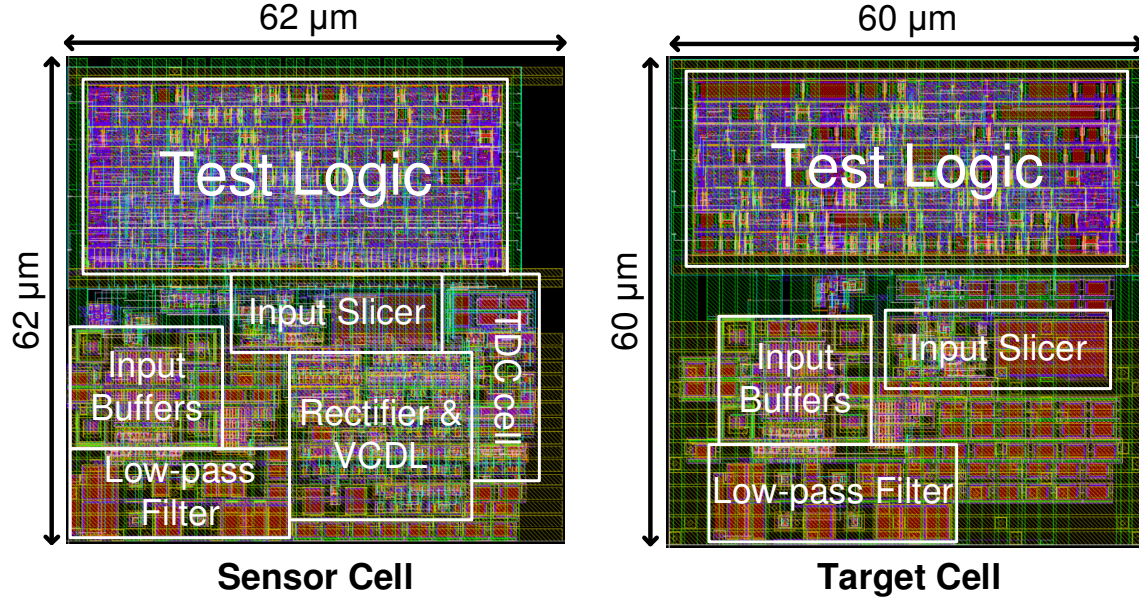


Figure 5.23: Sensor and target cell layout detail

Both a 6x4 cell (13x9 plate) sensor and 4x3 cell target array were implemented in the same 65 nm bulk CMOS test chip design (Figure 5.22). As described in section 5.1, target plate size is 60x60 μm; to accommodate $n = 2$ and a minimum pad-level metal spacing of 2 μm, sensor plates measure 29x29 μm. The sensor cell circuitry and associated test logic are designed to fit under a 2x2 set of sensor plates (total area of 62x62 μm) and connect by abutment to form the complete sensor array (Figure 5.23). The target cell electronics are likewise designed to fit under a single target plate; because the spacing between target plates is much larger than that between sensor plates (Figure 5.5), the target cells do not connect by abutment.

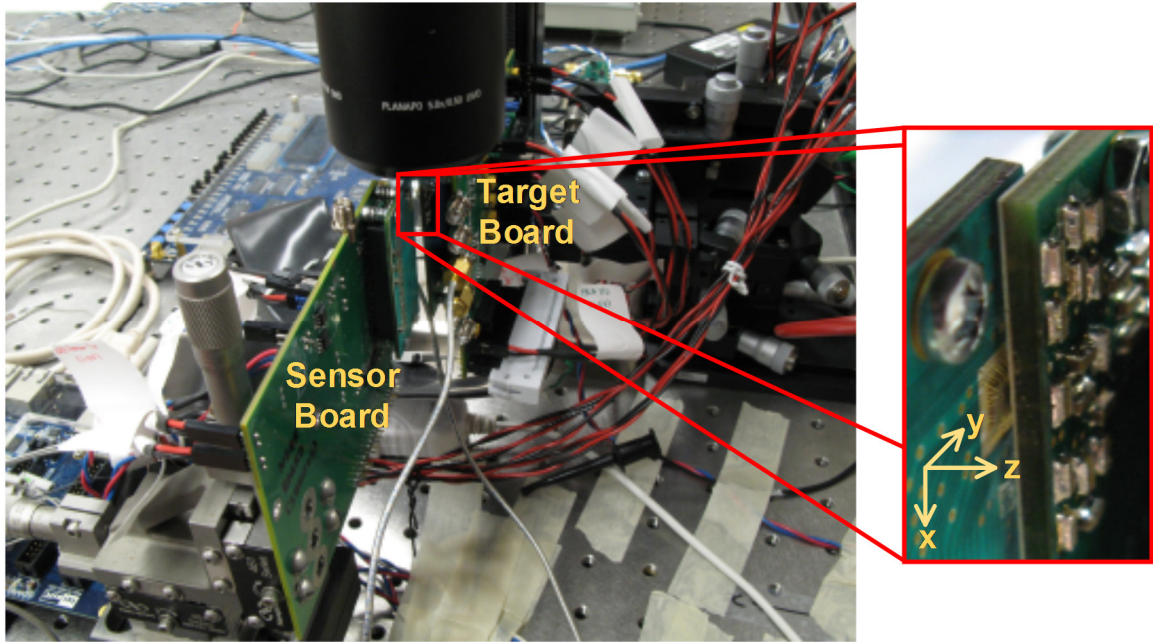


Figure 5.24: Test setup. Inset: Detail of chips when brought into alignment

The test setup (Figure 5.24) consists of two test chips mounted on small daughterboard PCBs using chip-on-board assembly. The daughterboards are used to provide sufficient clearance for access to connectors on the main PCB assembly, as well as to ease test chip replacement. The target chip mainboard is mounted on a 5-axis micropositioner, used to planarize the two chips in the θ_x - and θ_y -axes. Displacement of the two chips in the x -, y - and z -axes was controlled by means of the 5-axis micropositioner connected to the target board, as well as a 3-axis micropositioner connected to the sensor board. The θ_z -axis was left uncorrected due to limitations in the equipment available; this was mitigated by during assembly, by careful alignment of the chips against guides on the daughterboard. Initial alignment of the two chips was conducted visually using a microscope, with the alignment-sensing function of the sensor array used to correct any remaining alignment error. Slight over-torque was applied in order to cause the sensor and target boards to flex against each other and minimize the size of any air gaps between the chips or in the parylene dielectric (if present). Tests were conducted with different thicknesses of parylene (4, 5, 6, 2x4, 2x5 and 2x6 μm , with $\pm 0.5 \mu\text{m}$ tolerance per sheet) by placing a single sheet of parylene over the plate arrays on the sensor chip and fixing it to the daughterboard. A second sheet was added to the target board/chip as necessary.

The VCDL/TDC-based ADC was tested independently by setting the VCDL bias voltages through override pads, bypassing the input slicer and rectifier. To measure the effects of offset compensation, the transfer characteristic was measured both before and after the variable-threshold VCDL output buffers (Figure 5.10) were adjusted to minimize offset. An example of the effect of offset compensation is shown, for a single sensor cell, in Figure 5.25. Results were collected across all 24 sensor cells in 6 different chips (Figure 5.26), with an improvement in RMS offset error from 1.04 LSB to 0.38 LSB, and a corresponding improvement in RMS total error from 1.54 LSB to 0.97 LSB.

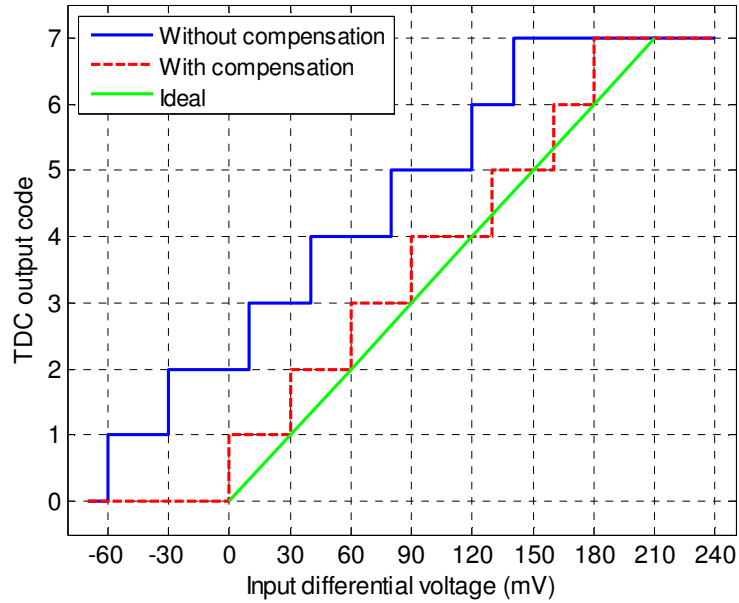


Figure 5.25: Effect of offset compensation on a single sensor cell's VCDL/TDC-based ADC

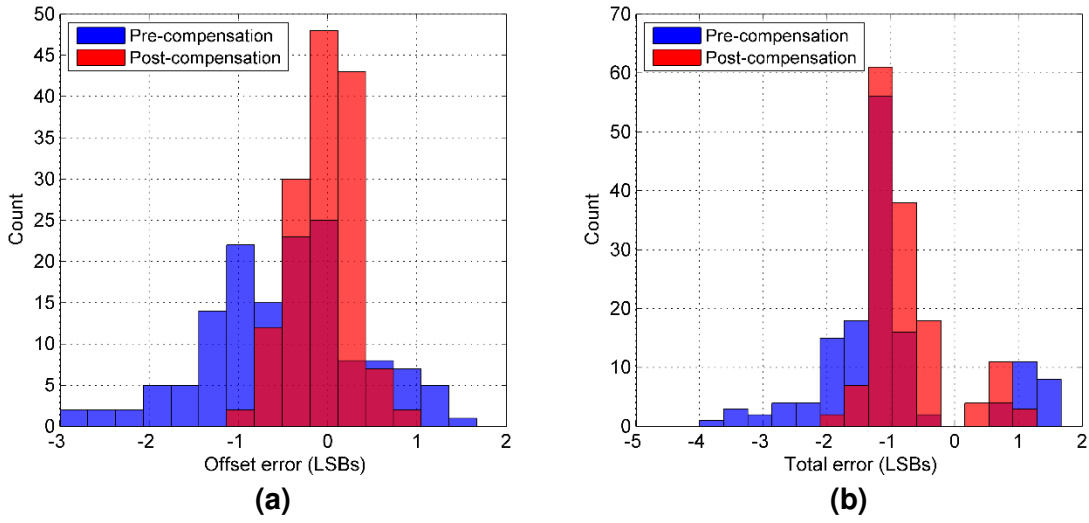


Figure 5.26: The effect of VCDL/TDC offset compensation on (a) offset error and (b) total error, measured over 144 sensor cells across 6 chips

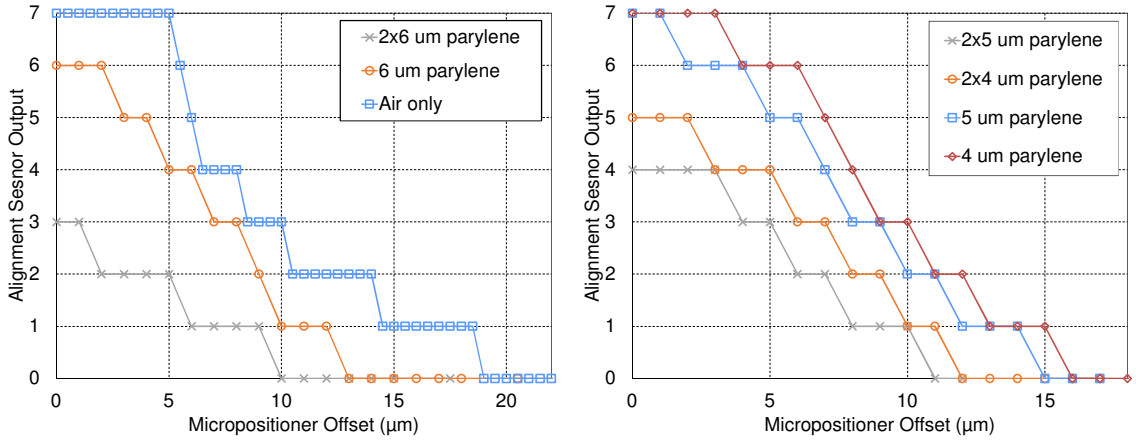


Figure 5.27: Alignment sensor output under vertical (z-axis) separation

Vertical (z-axis) alignment sensitivity was tested at the thicknesses of parylene listed above, as well as with an air-only dielectric (Figure 5.27). Readings for micropositioner offsets less than about 5 μm experience some non-linearity due to the over-torque applied to the chips, and coupling capacitances with the air-only and thinner (4 and 5 μm) parylene dielectrics experience strong enough coupling that the alignment sensor output saturates. Despite these non-idealities, it is clear from the vertical alignment tests that the sensor is performing largely as expected. When coupling is very strong (e.g. in the case of an air-only dielectric), the inverse dependence of

coupling strength on vertical separation results in a non-linear sensor output characteristic, essentially hyperbolic except where the sensor saturates. The introduction of a parylene dielectric limits coupling strength and effectively linearizes the sensor, but can result in significant degradation in sensitivity; sensor resolution is as poor as 4 $\mu\text{m}/\text{LSB}$ in the worst case, when a 2x6 μm parylene dielectric is used.

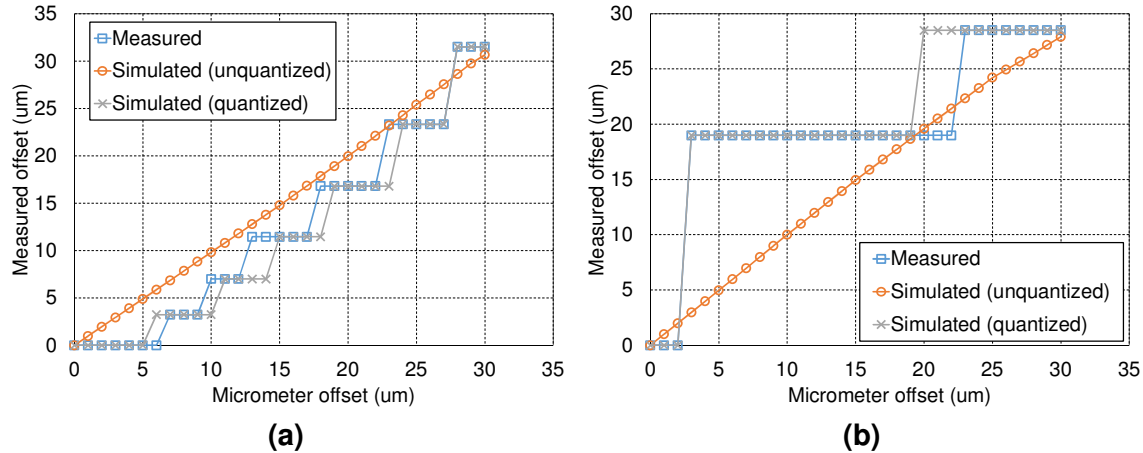


Figure 5.28: Comparison of measured and simulated (quantized and unquantized) in-plane (x- and y-axis) alignment sensing, with (a) air-only and (b) 2x6 μm parylene dielectrics

In theory, in-plane (x- and y-axis) alignment measurements should be insensitive to the presence of thicker parylene dielectrics, since they depend on a ratio between adjacent sensor outputs (Figure 5.14), both of which are equally affected by the reduction in coupling. In practice, because of the limited resolution of the ADC used, the reduction in coupling means a loss in alignment sensor resolution. For example, the simulation results of Figure 5.15, which was generated using analog output values, can be quantized using an ideal 3-bit ADC and compared with the measured results at the two extremes tested: with an air-only dielectric and 2x6 μm parylene dielectric (Figure 5.28). Resolution with no parylene and strong coupling is about 4 μm , degrading to 19 μm when the thickest parylene dielectric is used and coupling is at its weakest. Results for the other thicknesses tested are presented in Figure 5.29, and show a similar trend of reduction in alignment resolution as the parylene thickness is increased (Figure 5.30); some discontinuity in the trend for 4 and 5 μm parylene thicknesses is likely a result of residual ADC error after offset compensation, especially because these results were collected using different sensor cells in different chips.

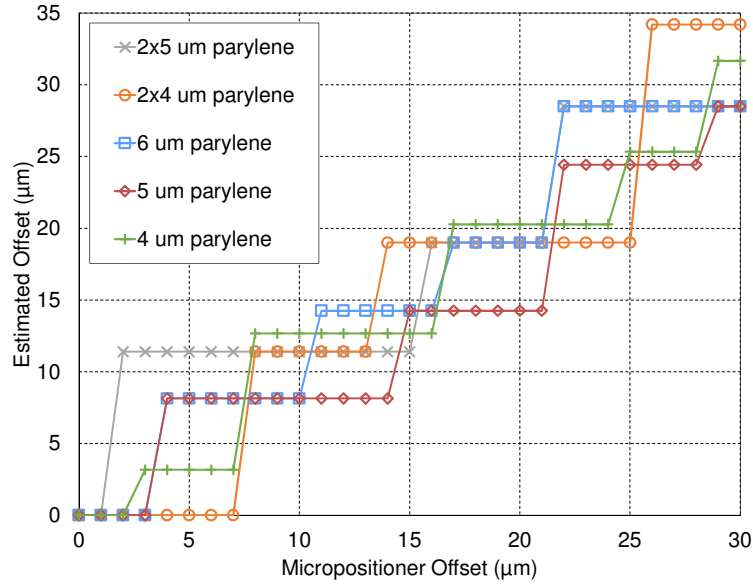


Figure 5.29: Alignment sensor output under in-plane (x- and y-axis) misalignment, for 4, 5, 6, 2x4 and 2x5 μm parylene dielectrics

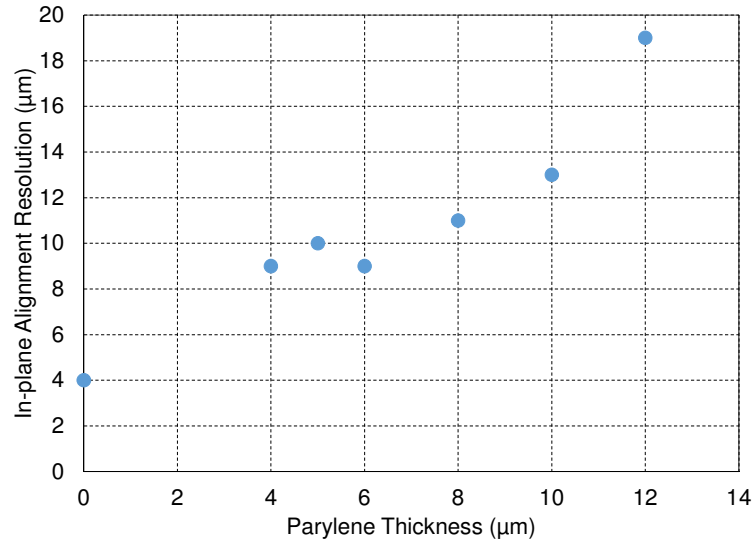


Figure 5.30: Achieved in-plane alignment sensor resolution vs. parylene dielectric thickness

When properly aligned, communication over all 12 available channels was demonstrated, at data rates up to 60 Mbps/channel with $\text{BER} < 10^{-9}$. Input slicer offset compensation was run every 2 ms, and given approximately 10 μs to converge (600 clock cycles @ 60 MHz), for a net

data rate of about 59.7 Mbps/channel, or 716 Mbps over all 12 channels. Maximum achievable data rate is dependent on the amount of coupling between target and sensor plates, and was measured across all available thicknesses of parylene (Figure 5.31).

Performance of the capacitive proximity interconnect is summarized in Table 5.2. This design achieves a power consumption figure-of-merit of 0.180 pJ/bit, competitive with the best-reported capacitive [100] and inductive [101] proximity interconnects to date, both of which achieve 0.14 pJ/bit, but do not include alignment sensing capability and are unidirectional, able to communicate only in a single direction from one chip to the other. Power consumed when the chips are used in alignment-sensing mode is minimal – only 6.4 μ W higher than idle power. The bulk of idle power consumption (about 16 μ W in each chip) is the global current tree, which is always active.

Process	65 nm bulk CMOS
Die Size	1.6 mm x 2.4 mm
Sensor Array Size	401 μ m x 277 μ m
Target Array Size	370 μ m x 267 μ m
Data Rate	12 x 60 Mbps
Transmitter & Input Buffer Supply	1.0 V
Slicer, Rectifier, VCDL & TDC Supply	0.7 V
Power Dissipation (Sensor + Target)	
Transceiver @ 12 x 60 Mbps	100.9 μ W + 27.7 μ W
Alignment Sensor	23.1 μ W + 20.2 μ W
Idle	18.0 μ W + 18.9 μ W
Figures-of-Merit	
Power	0.180 pJ/bit
Area (based on sensor array size)	6446 Mbps/mm ²

Table 5.2: Performance summary, 12 x 60 Mbps capacitive proximity interconnect with embedded alignment sensor

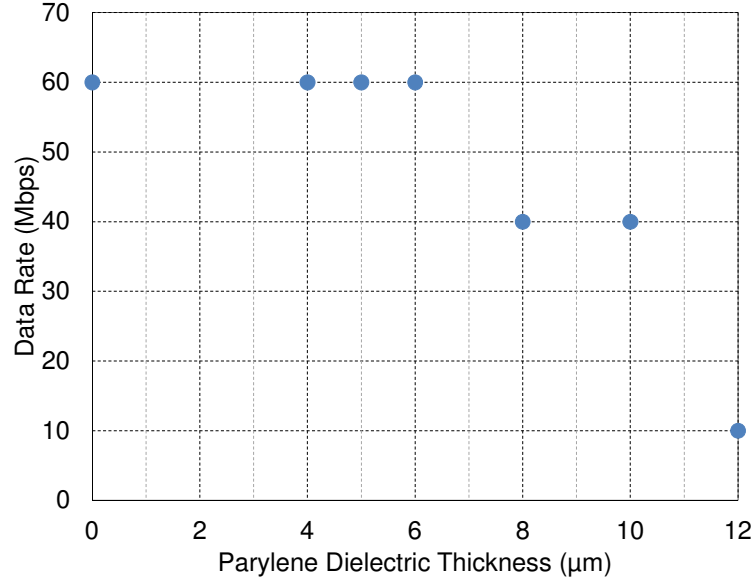


Figure 5.31: Maximum data rates achievable ($\text{BER} < 10^{-9}$) under best-case alignment, for various thickness of parylene dielectric

5.4 Summary

By integrating implant electronics onto a foldable parylene structure, the origami approach to design gives engineers a tool for overcoming the size, power and cost constraints typically faced when building a biomedical implant. With proper design, origami implants can even be used to realize mechanically useful shapes, thereby enhancing implant performance or providing new capabilities. Moving forward, the origami design style can also be extended to encompass a modular approach to implant design that envisions the development a library of standard functional blocks built using ICs in parylene, which can be assembled on-demand for custom implants.

A vital component of this vision is a wireless link that allows the various ICs in an origami implant to communicate with each other reliably and efficiently. A capacitive proximity interconnect for this purpose has been developed and fabricated in 65 nm CMOS. It contains an embedded alignment sensor, which allows link quality to be assessed and the array adapted so that only the best-coupled (and therefore most power efficient) links are used whenever possible. This ensures robustness and reliability in the face of misalignment due to fabrication tolerances,

patient movement or other perturbations. By obviating the need for a separate alignment sensor, the embedded sensor saves area and simplifies the array adaptation logic.

The alignment sensor uses a rectifier that is controlled by the receiver's input slicer; this re-use of existing hardware makes the design more compact and reduces leakage power. To further save area and power, the sensor's ADC is formed from a TDC that is distributed across the transceiver array. Transceiver itself has been optimized for power-efficient communication at the upper end of typical biomedical data rates, in the 10-60 Mbps range, and this has been demonstrated through dielectrics as thick as 12 μm of parylene-C with a power consumption of 0.180 pJ/bit.

Chapter 6: Conclusion

Driven by the performance, power and size benefits derived from CMOS scaling, integrated electronics have found themselves pushing the envelopes not only of raw performance, but also ubiquity. As more and more performance and features are demanded of ICs in both the high and low extremes of the power, performance and size spectra, traditional approaches to the challenges of chip-to-chip interconnect have rapidly run out of steam. In the high-performance space, channel loss and distortion, headroom constraints, process variability and power consumption concerns have made conventional, analog-heavy designs unacceptably costly. On the other extreme, emerging applications such as biomedical implants and ubiquitous computing frequently require communication in environments less controlled and more hostile than traditional chip-to-chip interconnect design is suited for. Constraints on power and size drive innovative approaches to problems of reliability and robustness in such unconventional settings. Designs at both ends of the spectrum draw from a common pool of ideas and philosophies in order to address the challenges described above. Among these:

1. A heavy reliance on digital control and processing, leveraging the ever-diminishing marginal cost of transistors used as digital switches to overcome (or even entirely avoid) the high variability and area/power overheads of analog devices in highly-scaled CMOS.
2. Time-sharing of hardware across multiple data pins, thus minimizing the amount of idle hardware at any given time (conversely, this could be seen as maximizing the utilization of existing hardware), saving area and reducing leakage power, at some cost in performance.
3. Sharing hardware between different functional blocks within a single data pin, such as between CDR and adaptive equalizer, or between receiver and alignment sensor, also to reduce area and leakage.

4. Adaptive hardware, which is aware of required performance, senses actual link conditions and modifies the link(s) accordingly, to maximize power efficiency.

The ping-pong CDR uses an all-digital approach to capture the eye and determine the optimal data clock phase, thus avoiding the analog-heavy nature of the PLL/DLL-based designs traditionally required to realize infinite delay-range capability. A multi-pin extension of the algorithm shows how clocking and control hardware can be successfully shared between data pins in order to reduce power and area, albeit reducing the CDR bandwidth. The eye-monitor data produced by the CDR is useful for equalizer adaptation as well, and the adaptation algorithm presented describes how it can be used as such with minimal hardware overhead. The implemented multi-pin system is able to communicate at data rates between 3x6 Gb/s and 3x9 Gb/s, with power efficiency as good as 2.5 pJ/b, in 90 nm bulk CMOS.

The increasingly lossy nature of on-chip wires makes for a challenging link environment that encourages the adoption of synchronization and equalization techniques previously used only for board-level or longer interconnect. The all-digital nature of ping-pong CDR, its heavy reliance on synthesized, standard-cell-based control blocks, its amenability to sharing across multiple data pins, and its ability to perform both synchronization and equalization adaptation functions in the same set of hardware, particularly in the context of non-traditional DFE types such as IIR-DFE, make it a prime candidate for use in long-distance on-chip interconnect. Future work on the ping-pong CDR should develop these desirable characteristics, especially for large multi-core ICs, which increasingly rely on long on-chip wires to realize high-speed, low-latency communication for performance-critical functions such as cache coherency.

The capacitive proximity interconnect is a key enabler in the design of origami biomedical implants, which will allow inexpensive, modular 3D integration of implant electronics in mechanically useful shapes. To be successfully used in this context, the proximity interconnect needs to be extremely power-efficient, yet also able to sense its alignment and adapt accordingly. This is achieved by embedding an alignment sensor directly into the communication array, allowing straightforward determination of link coupling without resorting to complex look-up tables and/or inference calculations. To save power, a largely digital, TDC-based approach is used for the digitization of alignment information, and the TDC itself is distributed across

(shared between) array elements for compactness. Receiver and alignment sensor share much the same hardware (input buffers, low-pass filter and slicer), also to address concerns of size and leakage. The proximity interconnect was fabricated in 65 nm CMOS and tested with parylene dielectrics from 4 – 12 μm thickness. Data rates as high as 12x60 Mbps were achieved, with power efficiency up to 0.18 pJ/b. With the thickest (12 μm) parylene dielectric, alignment resolution was measured to be 4 μm in the z-axis and 19 μm in the x- and y-axes.

The concept of origami biomedical implants is one very much in its infancy, and much development work remains on the physical configuration of the folding and latching as well as the electronics to support it. For example, the most efficient way to wirelessly deliver power to such an implant, and then transmit it from module to module within the implant, is still an open question. Another worthy topic of investigation is the use of inductive, rather than capacitive, proximity interconnect, to take advantage of its better tolerance of misalignment between chips. The relatively poor power efficiency of inductive proximity interconnect is a challenge that will need to be tackled, as is the question of how to use it to conduct alignment sensing, which has not been done to date. Given the different power and range characteristics displayed by capacitive and inductive forms of proximity communication, it is possible (even likely) that some combination of the two styles is the optimal design approach – leveraging the better range of inductive coupling when two communicating chips cannot be placed face-to-face and need to be further apart, and using capacitive coupling where alignment is better controlled in order to save power. Consideration should also be given to the substrate used to fabricate the plates or coils used in the proximity interconnect. Integrating them onto the IC itself is appealing because it reduces pad count, limits parasitic loading and makes the system more compact, but the size of the plates or coils is limited by the relative expense of on-chip area, and they are subject to a thicker dielectric stack (parylene and chip passivation). If a smaller number of capacitive/inductive elements is required, integrating the plates/coils into the parylene itself might be desirable.

List of Abbreviations

ADC	Analog-to-digital converter
AMD	Age-related macular degeneration
AWGN	Additive white Gaussian noise
BER	Bit error rate
CDR	Clock-data recovery (system)
CID	Consecutive identical digits
CMOS	Complementary metal-oxide-semiconductor
CPU	Central processing unit
DCDL	Digitally-controlled delay line
DCO	Digitally-controlled oscillator
DFE	Decision-feedback equalizer
DLL	Delay-locked loop
DNL	Differential non-linearity
DRAM	Dynamic random-access memory
DVFS	Dynamic voltage and frequency scaling
Gb/s	Gigabit-per-second
GPU	Graphics processing unit
HDL	Hardware description language
HPC	High-performance computing
IC	Integrated circuit
I/O	Input/output
INL	Integral non-linearity
LPF	Low-pass filter
PCB	Printed circuit board
pJ/b	Picojoule-per-bit, a measure of link energy efficiency

PLL	Phase-locked loop
PRBS	Pseudo-random bit sequence (usually appended with a number indicating its length; for example, a PRBS-7 is $2^7-1 = 127$ bits long)
PVT	Process, voltage, temperature
RP	Retinitis pigmentosa
SJ	Sinusoidal jitter
SNR	Signal-to-noise ratio
SoC	System-on-chip
SS-LMS	Sign-sign least-mean-squares
TDC	Time-to-digital converter
PVT	Process, voltage and temperature
UI	Unit interval (one bit-time in a serial data stream)
VCDL	Voltage-controlled delay line
VCO	Voltage-controlled oscillator

Bibliography

- [1] G. Moore, "Cramming more components onto integrated circuits," *Electronics Magazine*, vol. 38, no. 8, Apr 1965.
- [2] L. Buechley and M. Eisenberg, "The LilyPad Arduino: Toward Wearable Engineering for Everyone," *IEEE Pervasive Computing*, vol. 7, no. 2, pp. 12-15, Apr-Jun 2008.
- [3] "MakerBot," [Online]. Available: <http://www.makerbot.com/>. [Accessed 18 Jun 2012].
- [4] S. Rusu, S. Tam, H. Muljono, J. Stinson, D. Ayers, J. Chang, R. Varada, M. Ratta, S. Kottapalli and S. Vora, "A 45 nm 8-Core Enterprise Xeon® Processor," *IEEE Journal Solid-State Circuits*, vol. 45, no. 1, pp. 7-14, Jan 2010.
- [5] F. O'Mahony, G. Balamurugan, J. E. Jaussi, J. Kennedy, M. Mansuri, S. Shekhar and B. Casper, "The Future of Electrical I/O for Microprocessors," in *IEEE Int. Symp. VLSI Design, Automation and Test (VLSI-DAT)*, 2009.
- [6] A. Majumdar, J. E. Cunningham and A. V. Krishnamoorthy, "Alignment and Performance Considerations for Capacitive, Inductive, and Optical Proximity Communication," *IEEE Transactions on Advanced Packaging*, vol. 33, no. 3, pp. 690-701, Aug 2010.
- [7] N. Miura, D. Mizoguchi, M. Inoue, T. Sakurai and T. Kuroda, "A 195-Gb/s 1.2-W Inductive Inter-Chip Wireless Superconnect With Transmit Power Control Scheme for 3-D-Stacked System in a Package," *IEEE Journal Solid-State Circuits*, vol. 41, no. 1, pp. 23-34, Jan 2006.
- [8] N. Miura, M. Saito and T. Kuroda, "A 1 TB/s 1 pJ/b 6.4 mm²/TB/s QDR Inductive-Coupling Interface Between 65-nm CMOS Logic and Emulated 100-nm DRAM," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems (JETCAS)*, vol. 2, no. 2, pp. 249-256, Jun 2012.
- [9] R. Drost, R. D. Hopkins, R. Ho and I. E. Sutherland, "Proximity Communication," *IEEE Journal Solid-State Circuits*, vol. 39, no. 9, pp. 1529-1535, Sep 2004.
- [10] D. Hopkins, A. Chow, R. Bosnyak, B. Coates, J. Edergen, S. Fairbanks, J. Gainsley, R. Ho, J. Lexau, F. Liu, T. Ono, J. Schauer, I. Sutherland and R. Drost, "Circuit Techniques to Enable 430Gb/s/mm² Proximity Communication," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, 2007.
- [11] M. M. Kim, M. Mehrara, M. Oskin and T. Austin, "Architectural Implications of Brick

- and Mortar Silicon Manufacturing," in *Int. Symp. on Computer Architecture (ISCA)*, 2007.
- [12] M. Javaheri, D. S. Hahn, R. R. Lakhanpal, J. D. Wiland and M. S. Humayun, "Retinal Prostheses for the Blind," *Annals of the Academy of Medicine Singapore*, vol. 35, no. 3, pp. 137-144, Mar 2006.
- [13] A. Y. Chow, V. Y. Chow, K. H. Packo, J. S. Pollack, G. A. Peyman and R. Schuchard, "The Artificial Silicon Retina microchip for the treatment of vision loss from retinitis pigmentosa," *Arch. Opthamol.*, vol. 122, no. 4, pp. 460-469, Apr 2004.
- [14] M. S. Humayun, J. D. Weiland, G. Y. Fujii, R. Greenberg, R. Williamson, J. Little, B. Mech, V. Cimarusti, G. Van Boemel, D. Gislin and E. de Juan Jr., "Visual perception in a blind subject with a chronic microelectronic retinal prosthesis," *Vision Research*, vol. 43, no. 24, pp. 2573-2581, Nov 2003.
- [15] K. Sooksood, E. Noorsal, J. Becker and M. Ortmanss, "A Neural Stimulator Front-End with Arbitrary Pulse Shape, HV Compliance and Adaptive Supply Requiring 0.05mm^2 in $0.35\mu\text{m}$ HVC MOS," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, 2011.
- [16] J. Chang, R. Huang and Y. C. Tai, "High-density IC chip integration with parylene pocket," in *IEEE Int. Conf. on NEMS*, Feb 2011.
- [17] Y. Zhao, M. S. Nandra and Y. C. Tai, "A MEMS Intraocular Origami Coil," in *Solid-State Sensors, Actuators and Microsystems Conference (TRANSDUCERS)*, 2011.
- [18] W. J. Dally and J. W. Poulton, *Digital Systems Engineering*, New York: Cambridge University Press, 2008, pp. 296-297.
- [19] A. C. Carusone, "An Equalizer Adaptation Algorithm to Reduce Jitter in Binary Receivers," *Circuits and Systems II: Express Briefs, IEEE Transactions on*, vol. 53, no. 9, pp. 807-811, Sep 2006.
- [20] JEDEC Standard, "DDR 3 SDRAM Specification," document no. JESD79-3E, Jul 2010.
- [21] HyperTransport Consortium, "HyperTransport™ I/O Link Specification," document no. HTC20051222-0046-0035, 2010.
- [22] N. Kurd, J. Douglas, P. Mosalikanti and R. Kumar, "Next generation Intel® micro-architecture (Nehalem) clocking architecture," in *IEEE Symp. VLSI Circuits Dig.*, Jun 2008.
- [23] E. Prete, D. Scheideler and A. Sanders, "A 100 mW 9.6 Gb/s transceiver in 90 nm CMOS for next-generation memory interfaces," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, 2006.

- [24] A. Bhatt, "Creating a PCI Express Interconnect," PCI-SIG White Paper, 2002.
- [25] C. R. Hogge Jr., "A self correcting clock recovery circuit," *Lightwave Technology, Journal of*, vol. 3, no. 6, pp. 1312-1314, Dec 1985.
- [26] H. Lee, A. Bansal, Y. Frans, J. Zerbe, S. Sidiropoulos and M. Horowitz, "Improving CDR Performance via Estimation," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, 2006.
- [27] M. Ramezani, M. Abdalla, A. Shoval, M. Van Ierssel, A. Rezayee, A. McLaren, C. Holdenried, J. Pham, E. So, D. Cassan and S. Sadr, "An 8.4mW/Gb/s 4-Lane 48Gb/s Multi-Standard-Compliant Transceiver in 40nm Digital CMOS Technology," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, 2011.
- [28] J. Alexander, "Clock recovery from random binary signals," *Electronics Letters*, vol. 11, no. 22, pp. 541-542, Dec 1975.
- [29] R. C. Walker, "Designing Bang-Bang PLLs for Clock and Data Recovery in Serial Data Transmission Systems," in *Phase-Locking in High-Performance Systems*, B. Razavi, Ed., Piscataway, NJ, IEEE Press, 2003, pp. 34-45.
- [30] K. H. Mueller and M. Müller, "Timing Recovery in Digital Synchronous Data Receivers," *IEEE Transactions on Communications*, vol. 24, no. 5, pp. 516-531, May 1976.
- [31] V. Balan, J. Caroselli, J.-G. Chern, C. Chow, R. Dadi, C. Desai, L. Fang, D. Hsu, P. Joshi, H. Kimura, C. Y. Liu, T.-W. Pan, R. Park, C. You, Y. Zeng, E. Zhang and F. Zhong, "A 4.8-6.4-Gb/s Serial Link for Backplane Applications Using Decision Feedback Equalization," *IEEE Journal Solid-State Circuits*, vol. 40, no. 9, pp. 1957-1967, Sep 2005.
- [32] F. Spagna, L. Chen, M. Deshpande, Y. Fan, D. Gambetta, S. Gowder, S. Iyer, R. Kumar, P. Kwok, R. Krishnamurthy, C.-c. Lin, R. Mohanavelu, R. Nicholson, J. Ou, M. Pasquarella, K. Prasad, H. Rustam, L. Tong, A. Tran, J. Wu and X. Zhang, "A 78mW 11.8Gb/s Serial Link Transceiver with Adaptive RX Equalization and Baud-Rate CDR in 32nm CMOS," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, 2010.
- [33] V. Stojanović, A. Ho, B. W. Garlepp, F. Chen, J. Wei, G. Tsang, E. Alon, R. T. Kollipara, C. W. Werner, J. L. Zerbe and M. A. Horowitz, "Autonomous Dual-Mode (PAM2/4) Serial Link Transceiver With Adaptive Equalization and Data Recovery," *IEEE Journal Solid State-Circuits*, vol. 40, no. 4, pp. 1012-1026, Apr 2005.
- [34] D. K. Weinlader, "Precision CMOS Receivers for VLSI Testing Applications," Ph.D. dissertation, Dept. Elect. Eng., Stanford University, Stanford, CA, 2001.

- [35] C.-K. K. Yang, "Delay-Locked Loops - An Overview," in *Phase Locking in High-Performance Systems*, B. Razavi, Ed., Piscataway, NJ, IEEE Press, 2003, pp. 13-22.
- [36] S. Sidiropoulos and M. Horowitz, "A Semidigital Dual Delay-Locked Loop," *IEEE Journal Solid-State Circuits*, vol. 32, no. 11, pp. 1683-1692, Nov 1997.
- [37] T. O. Dickson, J. F. Bulzacchelli and D. J. Friedman, "A 12-Gb/s 11-mW Half-Rate Sampled 5-Tap Decision Feedback Equalizer With Current-Integrating Summers in 45-nm SOI CMOS Technology," *IEEE Journal Solid-State Circuits*, vol. 44, no. 4, pp. 1298-1305, Apr 2009.
- [38] A. Ho, V. Stojanović, F. Chen, C. Werner, G. Tsang, E. Alon, R. Kollipara, J. Zerbe and M. Horowitz, "Common-mode Backchannel Signaling System for Differential High-speed Links," in *IEEE Symp. VLSI Circuits Dig.*, Jun 2004.
- [39] J. G. Proakis and M. Salehi, *Digital Communications*, 5th ed., New York: McGraw-Hill, 2008.
- [40] T. Beukema, M. Sorna, K. Selander, S. Zier, B. L. Ji, P. Murfet, J. Mason, W. Rhee, H. Ainspan, B. Parker and M. Beakes, "A 6.4-Gb/s CMOS SerDes Core With Feed-Forward and Decision-Feedback Equalization," *IEEE Journal Solid-State Circuits*, vol. 40, no. 12, pp. 2633-2645, Dec 2005.
- [41] B. Kim, Y. Liu, T. O. Dickson, J. F. Bulzacchelli and D. J. Friedman, "A 10-Gb/s Compact Low-Power Serial I/O With DFE-IIR Equalization in 65-nm CMOS," *IEEE Journal Solid-State Circuits*, vol. 44, no. 12, pp. 3526-3538, Dec 2009.
- [42] S. Kasturia and J. H. Winters, "Techniques for high-speed implementation of nonlinear cancellation," *IEEE Journal on Selected Areas in Communication*, vol. 9, no. 5, pp. 711-717, Jun 1991.
- [43] A. Emami-Neyestanak, A. Varzaghani, J. F. Bulzacchelli, A. Rylyakov, C.-K. K. Yang and D. J. Friedman, "A 6.0-mW 10.0-Gb/s Receiver With Switched-Capacitor Summation DFE," *IEEE J. Solid-State Circuits*, vol. 42, no. 4, pp. 889-896, Apr. 2007.
- [44] J. Sonntag and J. Stonick, "A digital clock and data recovery architecture for multi-gigabit/s binary links," *IEEE J. Solid-State Circuits*, vol. 41, no. 8, pp. 1867-1875, Aug 2006.
- [45] P. K. Hanumolu, M. G. Kim, G.-Y. Wei and U. Moon, "A 1.6 Gbps digital clock and data recovery circuit," in *Proc. IEEE Custom Integrated Circuits Conf. (CICC '06)*, Sep 2006.
- [46] M. Perrott, Y. Huang, R. Baird, B. Garlepp, D. Pastorello, E. King, Q. Yu, D. Kasha, P. Steiner, L. Zhang, J. Hein and B. Del Signore, "A 2.5-Gb/s multi-rate 0.25- μ m CMOS clock and data recovery circuit utilizing a hybrid analog/digital loop filter and

- all-digital referenceless frequency acquisition," *IEEE Journal Solid-State Circuits*, vol. 41, no. 12, pp. 2930-2944, Dec 2006.
- [47] J. Montanaro, R. Witek, K. Anne, A. Black, E. Cooper, D. Dobberpuhl, P. Donahue, J. Eno, W. Hoepfner, D. Kruckemyer, T. Lee, P. Lin, L. Madden, D. Murray, M. Pearce, S. Santhanam, K. Snyder, R. Stehpany and S. Thierauf, "A 160-MHz, 32-b, 0.5-W CMOS RISC microprocessor," *IEEE Journal Solid-State Circuits*, vol. 31, no. 11, pp. 1703-1714, Nov 1996.
- [48] Y. Tomita, M. Kibune, J. Ogawa, W. Walker, H. Tamura and T. Kuroda, "A 10-Gb/s receiver with series equalizer and on-chip ISI monitor in 0.11- μ CMOS," *IEEE Journal Solid-State Circuits*, vol. 40, no. 4, pp. 986-993, Apr 2005.
- [49] B. Analui, A. Rylyakov, S. Rylov, M. Meghelli and A. Hajimiri, "A 10-Gb/s two-dimensional eye-opening monitor in 0.13- μ m standard CMOS," *IEEE Journal Solid-State Circuits*, vol. 40, no. 12, pp. 2689-2699, Dec 2005.
- [50] E.-H. Chen, J. Ren, B. Leibowitz, H.-C. Lee, Q. Lin, K. Oh, F. Lambrecht, V. Stojanović, J. Zerbe and C.-K. Yang, "Near-optimal equalizer and timing adaptation for I/O links using a BER-based metric," *IEEE Journal Solid-State Circuits*, vol. 43, no. 9, pp. 2144-2156, Sep 2008.
- [51] T. Suttorp and U. Langmann, "A 10-Gb/s CMOS serial-link receiver using eye-opening monitoring for adaptive equalization and for clock and data recovery," in *Proc. IEEE Custom Integrated Circuits Conf. (CICC '07)*, Sep 2007.
- [52] H. Noguchi, N. Yoshida, H. Uchida, M. Ozaki, S. Kanemitsu and S. Wada, "A 50-Gb/s CDR circuit with adaptive decision-point control based on eye-monitor feedback," *IEEE Journal Solid-State Circuits*, vol. 43, no. 12, pp. 2929-2938, Dec 2008.
- [53] D. Oh, H. Lan, C. Madden, S. Chang, L. Yang and R. Schmitt, "In-situ characterization of 3D package systems with on-chip measurements," in *Proc. 60th Electronic Components and Technology Conf. (ECTC)*, Jun 2010.
- [54] M. Loh and A. Emami-Neyestanak, "All-digital CDR for high-density, high-speed I/O," in *IEEE Symp. VLSI Circuits Dig.*, Jun. 2010.
- [55] M. Loh and A. Emami-Neyestanak, "A 3x9 Gb/s Shared, All-Digital CDR for High-Speed, High-Density I/O," *IEEE Journal Solid-State Circuits*, vol. 47, no. 3, pp. 641-651, Mar 2012.
- [56] J. Tierno, A. Rylyakov and D. Friedman, "A wider power supply range, wide tuning range, all static CMOS all digital PLL in 65 nm SOI," *IEEE Journal Solid-State Circuits*, vol. 43, no. 1, pp. 42-51, Jan 2008.

- [57] S.-J. Lee, B. Kim and K. Lee, "A novel high-speed ring oscillator for multiphase clock generation using negative skewed delay scheme," *IEEE Journal Solid-State Circuits*, vol. 32, no. 2, pp. 289-291, Feb 1997.
- [58] N. H. E. Weste and D. Harris, *CMOS VLSI Design: A Circuits and Systems Perspective*, 3rd ed., Boston, MA: Pearson Education, 2005.
- [59] A. Nalamalpu, S. Srinivasan and W. Burleson, "Boosters for driving long on-chip interconnects: design issues, interconnect synthesis and comparison with repeaters," *IEEE Trans. Computer-Aided Design of Int. Cir. & Sys.*, vol. 21, no. 1, pp. 50-62, Jan 2002.
- [60] H. Kaul and D. Sylvester, "Low-power on-chip communication based on transition-aware global signaling (TAGS)," *IEEE Trans. VLSI Sys.*, vol. 12, no. 5, pp. 464-476, May 2004.
- [61] R. T. Chang, N. Talwalkar, C. P. Yue and S. S. Wong, "Near speed-of-light signaling over on-chip electrical interconnects," *IEEE Journal Solid-State Circuits*, vol. 38, no. 5, pp. 834-838, 2003.
- [62] A. P. Jose, G. Patounakis and K. L. Shepard, "Pulsed current-mode signaling for nearly speed-of-light intrachip communication," *IEEE Journal Solid-State Circuits*, vol. 41, no. 4, pp. 772-780, Apr 2006.
- [63] B. Kim and V. Stojanović, "An energy-efficient equalized transceiver for RC-dominant channels," *IEEE Journal Solid-State Circuits*, vol. 45, no. 6, pp. 1186-1197, Jun 2010.
- [64] R. Ho, T. Ono, R. D. Hopkins, A. Chow, J. Scahuer, F. Y. Liu and R. D. Drost, "High speed and low energy capacitively driven on-chip wires," *IEEE Journal Solid-State Circuits*, vol. 43, no. 1, pp. 52-60, Jan 2008.
- [65] S.-C. Wong, G.-Y. Lee and D.-J. Ma, "Modeling of Interconnect Capacitance, Delay, and Crosstalk in VLSI," *Semiconductor Manufacturing, IEEE Transactions on*, vol. 13, no. 1, pp. 108-111, Feb 2000.
- [66] E. Mensink, D. Schinkel, E. A. M. Klumperink, E. van Tuijl and B. Nauta, "Power efficient gigabit communication over capacitively driven RC-limited on-chip interconnects," *IEEE Journal Solid-State Circuits*, vol. 45, no. 2, pp. 447-457, Feb 2010.
- [67] Y.-C. Huang and S.-I. Liu, "A 6Gb/s receiver with 32.7dB adaptive DFE-IIR equalization," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, 2011.
- [68] T. A. Edison, "Means for transmitting signals electrically". United States of America

Patent 465971, 29 Dec 1891.

- [69] T. A. Edison, "The Air-Telegraph: System of Telegraphing to Trains and Ships," *The North American Review*, vol. 142, no. 352, pp. 285-291, Mar 1886.
- [70] J. J. Fahie, A History of Wireless Telegraphy, London: William Blackwood and Sons, 1901, pp. 100-111.
- [71] D. B. Salzman and T. F. Knight, Jr., "Capacitively Coupled Multichip Modules," in *Multi-Chip Module Conf. Proc.*, Apr. 1994.
- [72] S. A. Kuhn, M. B. Kleiner, R. Thewes and W. Weber, "Vertical Signal Transmission in Three-Dimensional Integrated Circuits by Capacitive Coupling," in *Proc. Int. Symp. on Circuits and Systems (ISCAS)*, May 1995.
- [73] K. Kanda, D. D. Antono, K. Ishida, H. Kawaguchi, T. Kuroda and T. Sakurai, "1.27Gb/s/pin 3mW/pin wireless superconnect (WSC) interface scheme," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2003.
- [74] A. V. Krishnamoorthy, J. E. Cunningham, X. Zheng, I. Shubin, J. Simons, D. Feng, H. Liang, C.-C. Kung and M. Asgari, "Optical Proximity Communication with Passively Aligned Silicon Photonic Chips," *IEEE J. Quantum Electron.*, vol. 45, no. 4, pp. 409-414, Apr. 2009.
- [75] A. Fazzi, L. Magagni, M. Mirandola, B. Charlet, L. Di Cioccio, E. Jung, R. Canegallo and R. Guerrieri, "3-D Capacitive Interconnections for Wafer-Level and Die-Level Assembly," *IEE J. Solid-State Circuits*, vol. 42, no. 10, pp. 2270-2282, Oct. 2007.
- [76] A. Fazzi, R. Canegallo, L. Ciccarelli, L. Magagni, F. Natali, E. Jung, P. Rolandi and R. Guerrieri, "3-D Capacitive Interconnections With Mono- and Bi-Directional Capabilities," *IEEE J. Solid-State Circuits*, vol. 43, no. 1, pp. 275-284, Jan. 2008.
- [77] J. E. Cunningham, A. V. Krishnamoorthy, I. Shubin, X. Zheng, M. Asghari, D. Feng and J. G. Mitchell, "Aligning Chips Face-to-Face for Dense Capacitive and Optical Communication," *IEEE Adv. Packag.*, vol. 33, no. 2, pp. 389-397, May 2010.
- [78] A. Chow, D. Hopkins, R. Ho and R. Drost, "Measuring 6D Chip Alignment in Multi-Chip Packages," in *IEEE Sensors*, Oct. 2007.
- [79] R. Canegallo, M. Mirandola, A. Fazzi, L. Magagni, R. Guerrieri and K. Kaschlun, "Electrical Measurement of Alignment for 3D Stacked Chips," in *Proc. European Solid-Static Circuits Conf. (ESSCIRC)*, Sep. 2005.
- [80] Y.-S. Lin, D. Sylvester and D. Blaauw, "Alignment-Independent Chip-to-Chip Communication for Sensor Applications Using Passive Capacitive Signaling," *IEEE J. Solid-State Circuits*, vol. 44, no. 4, pp. 1156-1166, Apr. 2009.

- [81] R. Drost, R. Ho, D. Hopkins and I. Sutherland, "Electronic Alignment for Proximity Communication," in *Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2004.
- [82] N. Miura, D. Mizoguchi, T. Sakurai and T. Kuroda, "Analysis and Design of Inductive Coupling and Transceiver Circuit for Inductive Inter-Chip Wireless Superconnect," *IEEE J. Solid-State Circuits*, vol. 40, no. 4, pp. 829-837, Apr. 2005.
- [83] C. R. Paul, *Inductance: Loop and Partial*, Hoboken, NJ: John Wiley & Sons, Inc., 2010.
- [84] N. Miura, Y. Kohama, Y. Sugimori, H. Ishikuro, T. Sakurai and T. Kuroda, "A High-Speed Inductive-Coupling Link with Burst Transmission," *IEEE J. Solid-State Circuits*, vol. 44, no. 3, pp. 947-955, Mar. 2009.
- [85] R. F. Yazicioglu, S. Kim, T. Torfs, P. Merken and C. Van Hoof, "A 30 μ W Analog Signal Processor ASIC for biomedical signal monitoring," in *Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2010.
- [86] X. Zou, W.-S. Liew, L. Yao and Y. Lian, "A 1V 22 μ W 32-channel implantable EEG recording IC," in *Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2010.
- [87] N. Van Helleputte, S. Kim, H. Kim, J. P. Kim, C. Van Hoof and R. F. Yazicioglu, "A 160 μ A biopotential acquisition ASIC with fully integrated IA and motion-artifact suppression," in *Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2012.
- [88] C. M. Lopez, A. Andrei, S. Mitra, M. Welkenhuysen, W. Eberle, C. Bartic, R. Puers, R. F. Yazicioglu and G. Gielen, "An Implantable 455-Active-Electrode 52-Channel CMOS Neural Probe," in *Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2013.
- [89] D. Han, Y. Zheng, R. Rajkumar, G. Dawe and M. Je, "A 0.45V 100-Channel Neural-Recording IC with Sub- μ W/Channel Consumption in 0.18 μ m CMOS," in *Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2013.
- [90] H. Park, B. Gosselin, M. Kiani, H.-M. Lee, J. Kim, X. Huo and M. Ghovanloo, "A Wireless Magnetoresistive Sensing System for an Intra-Oral Tongue-Computer Interface," in *Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2012.
- [91] K. Chen, Y.-K. Lo and W. Liu, "A 37.6mm² 1024-Channel High-Compliance-Voltage SoC for Epiretinal Prostheses," in *Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2013.
- [92] M. Monge, M. Raj, M. Honarvar-Nazari, H.-C. Chang, Y. Zhao, J. Weiland, M. Humayun, Y.-C. Tai and A. Emami-Neyestanek, "A Fully Intraocular 0.0169mm²/pixel 512-Channel Self-Calibrating Epiretinal Prosthesis in 65nm CMOS," in *Int. Solid-State*

- Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2013.
- [93] A. Yakovlev, D. Pivonka, T. Meng and A. Poon, "A mm-sized wirelessly powered and remotely controlled locomotive implantable device," in *Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2012.
 - [94] E. Y. Chow, S. Chakraborty, W. J. Chappell and P. P. Irazoqui, "Mixed-signal Integrated Circuits for Self-Contained Sub-Cubic Millimeter Biomedical Implants," in *Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2010.
 - [95] S. Lange, H. Xu, C. Lang, H. Pless, J. Becker, H.-J. Tiedkte, E. Hennig and M. Ortmanns, "An AC-Powered Optical Receiver Consuming 270 μ W for Transcutaneous 2Mb/s Data Transfer," in *Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2011.
 - [96] G. K. Chen, H. Ghaed, R.-u. Haque, M. Wieckowski, Y. Kim, G. Kim, D. A. Fick, D. Kim, M. Seok, K. Wise, D. Blaauw and D. Sylvester, "A cubic-millimeter energy-autonomous wireless intraocular pressure monitor," in *Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2011.
 - [97] S. Lee, L. Yan, T. Roh, S. Hong and H.-J. Yoo, "A 75 μ W real-time scalable network controller and a 25 μ W ExG sensor IC for compact sleep-monitoring applications," in *Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2011.
 - [98] L. Yan, J. Bae, S. Lee, B. Kim, T. Roh, K. Song and H.-J. Yoo, "A 3.9mW 25-Electrode Reconfigured Thoracic Impedance/ECG SoC with Body-Channel Transponder," in *Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2010.
 - [99] P. Dudek, S. Szczepański and J. V. Hatfield, "A High-Resolution CMOS Time-to-Digital Converter Utilizing a Vernier Delay Line," *IEEE Journal Solid-State Circuits*, vol. 35, no. 2, pp. 240-247, Feb. 2000.
 - [100] A. Fazzi, L. Magagni, M. Mirandola, R. Canegallo, S. Schmitz and R. Guerrieri, "A 0.14mW/Gbps High-Density Capacitive Interface for 3D System Integration," in *IEEE Custom Integrated Circuits Conf. (CICC '05)*, Sep. 2005.
 - [101] N. Miura, H. Ishikuro, K. Niitsu, T. Sakurai and T. Kuroda, "A 0.14 pJ/b Inductive-Coupling Transceiver With Digitally-Controlled Precise Pulse Shaping," *IEEE J. Solid-State Circuits*, pp. 285-291, Jan 2008.